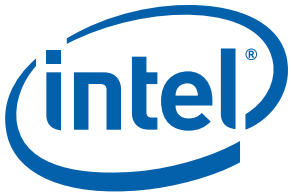


# Intel Atom® Processor C3000 Product Family

## Datasheet

***February 2018***

Order Number: 337018-002

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, the Intel logo, Pentium, Celeron, Atom, Core, Xeon and others are trademarks of Intel Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

*Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All rights reserved.

# Revision History

| Date | Revision | Description |
|---|---|---|
| February 2018 | 002US | Below is an overview of broad changes:<br>Chapter 31, "Intel® QuickAssist Technology (Intel® QAT)"<br>• Updated Intel® QuickAssist Reference.<br>**Included Register Chapters from the Intel Atom® Processor C3000 Product Family External Design Specification Volume 4 into this Datasheet.** |
| January 2018 | 001US | Initial release. |

# Contents

**Volume 1: Program Overview** ..... 105

**1 Introduction and Product Offerings** ..... **106**
1.1 Overview ..... 106
1.2 SoC Key Features ..... 113
1.2.1 Processor Details ..... 114
1.2.2 Supported Technologies ..... 114
1.2.3 System Memory Support ..... 115
1.2.4 Integrated Clock Generator ..... 116
1.2.5 Intel® QuickAssist Technology (Intel® QAT) ..... 116
1.2.6 Intel® Management Engine (Intel® ME) ..... 117
1.2.7 Innovation Engine (IE) ..... 117
1.2.8 Interface Features ..... 118
1.2.8.1 High Speed I/O Lanes ..... 118
1.2.8.2 LAN Ports ..... 119
1.2.8.3 SMBus Controllers ..... 119
1.2.8.4 Embedded Multi Media Card (eMMC) ..... 119
1.2.8.5 High-Speed Universal Asynchronous Receiver/Transmitter ..... 119
1.2.8.6 Low Pin Count (LPC) ..... 120
1.2.8.7 Serial Peripheral Interface (SPI) ..... 120
1.2.9 Package ..... 120
1.2.10 Testability ..... 120
1.3 Terminology ..... 121
1.4 Related Documents ..... 127

**Volume 2: Functional** ..... 130

**2 Multi-Core Intel Atom® Processors** ..... **131**
2.1 Introduction ..... 131
2.2 Features ..... 132
2.2.1 Goldmont New Technologies and ISA Extensions ..... 133
2.2.2 Intel® Virtualization Technology ..... 134
2.2.3 Intel® VT-x Objectives ..... 134
2.2.4 Intel® VT-x Features ..... 135
2.2.5 Security and Cryptography Technologies ..... 136
2.2.5.1 Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI) ..... 136
2.2.5.2 Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions) ..... 137
2.2.5.3 PCLMULQDQ Instruction ..... 137
2.2.5.4 Digital Random Number Generator ..... 137
2.2.6 Intel® Turbo Boost Technology ..... 138
2.2.6.1 Intel® Turbo Boost Technology Frequency ..... 138
2.2.7 CPUID Instruction ..... 138
2.2.7.1 CPUID Leaf 0 — Basic CPUID Information ..... 140
2.2.7.2 CPUID Leaf 1 — Version and Feature Information ..... 141
2.2.7.3 CPUID Leaf 2 — Cache and TLB Information ..... 146
2.2.7.4 CPUID Leaf 3— Reserved ..... 146
2.2.7.5 CPUID Leaf 4— Deterministic Cache Parameters ..... 147
2.2.7.6 CPUID Leaf 5— MONITOR/MWAIT ..... 149
2.2.7.7 CPUID Leaf 6— Digital Thermometer and Power Management ..... 150
2.2.7.8 CPUID Leaf 7— Extended feature Flags ..... 151
2.2.7.9 CPUID Leaf 8 — Reserved ..... 153
2.2.7.10 CPUID Leaf 9 — Direct Cache Access (DCA) Information ..... 153

2.2.7.11 CPUID Leaf Ah — Architectural Performance Monitoring ..... 153
2.2.7.12 CPUID Leaf Bh — Extended Topology Enumeration ..... 154
2.2.7.13 CPUID Leaf Ch — Reserved ..... 154
2.2.7.14 CPUID Leaf Dh — Intel® Memory Protection Extensions (Intel® MPX), XSAVE Feature ..... 154
2.2.7.15 CPUID Leaf Eh — Reserved ..... 157
2.2.7.16 CPUID Leaf Fh — Platform and Cache Quality of Service (QoS) ..... 157
2.2.7.17 CPUID Leaf 10h — Platform and Cache QoS Enforcement Enumeration ..... 157
2.2.7.18 CPUID Leaf 11h — Reserved ..... 158
2.2.7.19 CPUID Leaf 12h — Reserved ..... 158
2.2.7.20 CPUID Leaf 13h — Reserved ..... 158
2.2.7.21 CPUID Leaf 14h — Intel® Processor Trace (Intel® PT) Enumeration ..... 159
2.2.7.22 CPUID Leaf 15h — Time Stamp Counter (TSC) and Crystal Clock Ratio ..... 160
2.2.7.23 CPUID Leaves 16h through 7FFFFFFFh — Reserved ..... 161
2.2.7.24 CPUID Leaf 80000000h — Maximum EAX Value for CPUID Instruction ..... 161
2.2.7.25 CPUID Leaf 80000001h — Extended Feature Flags ..... 162
2.2.7.26 CPUID Leaves 80000002h, 80000003h, and 80000004h— Intel Processor Brand String ..... 163
2.2.7.27 CPUID Leaf 80000005h — Reserved ..... 163
2.2.7.28 CPUID Leaf 80000006h — Cache Parameters ..... 163
2.2.7.29 CPUID Leaf 80000007h — Advanced Power Management ..... 164
2.2.7.30 CPUID Leaf 80000008h — Virtual/Physical Address Sizes ..... 164

**3 System Address Map ..... 165**
3.1 Host Root Space—Memory Space ..... 166
3.1.1 Host Memory Space—Regions and Boundaries ..... 166
3.1.2 Host Memory Space—Relocatable Memory-Mapped I/O ..... 169
3.2 Host Root Space—I/O Space ..... 174
3.2.1 Host I/O Space—Fixed I/O Addresses ..... 175
3.2.2 Host I/O Space—Relocatable I/O Addresses ..... 180
3.3 Host Root Space—PCI Configuration Register Space ..... 182
3.3.1 How to Access Registers in PCI Configuration Space ..... 185
3.4 Host Root Space - Host Sideband Register Access ..... 187
3.4.1 Using SBREG_BAR Method ..... 187
3.4.2 Accessing System Agent Sideband Registers ..... 187
3.5 Key Registers Mentioned in this Chapter ..... 188
3.5.1 Key Registers in Host Configuration Space ..... 188
3.5.2 Key Registers in Host Memory Space ..... 190
3.5.3 Key Registers in Host I/O Space ..... 190

**4 Strapping and Configuration ..... 191**
4.1 SoC Hard (Pin) Straps ..... 193
4.2 SoC Soft Straps ..... 199
4.3 Fuses Pertaining to the Innovation Engine (IE) ..... 260

**5 System Agent and Root Complex ..... 261**
5.1 Introduction ..... 261
5.2 Features ..... 262
5.3 Root Complex ..... 262
5.4 Register Map ..... 263
5.4.1 Registers in Configuration Space ..... 264

**6 Reliability, Availability and Serviceability (RAS) ..... 265**
6.1 SoC RAS Overview ..... 265

6.2 Signal Descriptions ..... 265
6.3 Feature Summary ..... 266
6.4 Architectural Overview ..... 267
6.4.1 Error Classification ..... 267
6.4.2 Error Detection and Reporting Mechanisms ..... 268
6.5 Availability Features ..... 269
6.5.1 System Memory ..... 269
6.5.1.1 Single Bit Error Correction ..... 269
6.5.1.2 DRAM Data Scrubbing ..... 269
6.5.1.3 DRAM Data Scrambling ..... 269
6.5.2 PCI Express Root Ports ..... 270
6.6 Machine Check Architecture (MCA) ..... 271
6.6.1 Machine Check MSR Addresses ..... 272
6.6.2 Enhancements to MCA Error Reporting ..... 274
6.6.2.1 Error Cloaking ..... 274
6.6.2.2 Dual Signaling ..... 274
6.6.2.3 Corrected Machine-Check error Interrupt (CMCI) ..... 275
6.7 Advanced Error Reporting (AER) ..... 276
6.7.1 AER and the PCIe Root Ports and RCEC ..... 276
6.8 Intel® Virtualization Technology (Intel® VT) ..... 277
6.9 Error Injection ..... 277
6.10 SoC UnCore and I/O Error Handling ..... 278
6.11 Hardware Error Classification and System Events ..... 279
6.11.1 Correctable Errors ..... 279
6.11.2 Uncorrectable Errors ..... 280
6.11.2.1 Fatal Errors ..... 280
6.11.2.2 Non-Fatal Errors ..... 280
6.11.3 Software Correctable Errors ..... 280
6.12 SoC UnCore and I/O Error Reporting ..... 281
6.12.1 Reporting Error to an External Device ..... 283
6.12.2 PCI Express INTx and MSI ..... 283
6.13 Overview of SoC Error Registers ..... 284
6.13.1 Local Error Registers ..... 285
6.13.2 SoC Global Error Registers ..... 286
6.13.3 First Error and Next Error Log Registers ..... 287
6.13.4 General Local and Global Error-Register Flow ..... 288
6.13.5 Global System Event Reporting ..... 290
6.13.6 Error Counters ..... 292
6.13.6.1 Error Counter Requirements ..... 292
6.14 Error Reporting Flow — PCI Express Devices/Functions ..... 293
6.14.1 Upstream Flow of Error Signal Messages ..... 294
6.14.2 How a PCIe Device Determines Which Error Message to Send ..... 295
6.14.3 PCIe Endpoint Devices Behind Virtual Root Port Error Handling ..... 297
6.14.4 PCI Express Root Ports ..... 298
6.14.4.1 Unsupported Transactions and Unexpected Completions ..... 298
6.14.4.2 Unconnected Ports ..... 298
6.14.5 Internal Errors — SMBus-Host Controller ..... 298
6.14.6 Local Error Registers ..... 299
6.15 Error Reporting Flow — Legacy PCI Devices/Functions ..... 302
6.15.1 Local Error Registers ..... 302
6.16 Error Reporting Flow — Non-PCI devices ..... 304
6.16.1 Local Error Registers ..... 305
6.16.2 Error Injection Control and Registers — Memory Controller ..... 308
6.17 Error Reporting Flow — I/O Fabric ..... 309
6.17.1 Local Error Registers ..... 309

6.18 SoC Error Handling Summary ..... 309
6.19 Register Map ..... 320
6.19.1 Registers in Configuration Space—GLREG ..... 321
6.19.2 Registers in Configuration Space—RCEC ..... 323

**7 Clock Architecture ..... 325**
7.1 Signal Descriptions ..... 326
7.2 Feature List ..... 328
7.3 Architectural Overview ..... 329
7.3.1 ISCLK ..... 330
7.3.2 Real-Time Clock ..... 330
7.3.3 PMU SUS Clock ..... 330
7.3.4 Flex Clocks ..... 330
7.3.5 SVID Clock ..... 331
7.3.6 SATA Serial GPIO (SGPIO) Clock ..... 331
7.3.7 EMMC Clock ..... 331
7.3.8 DDR Clocks ..... 331
7.3.9 PCI Express Reference Clocks ..... 331
7.3.10 SPI Clock ..... 331
7.3.11 LPC Clocks ..... 331
7.3.12 LAN I2C Clocks ..... 331
7.3.13 LAN MDC Clock ..... 332
7.3.14 LAN NC-SI Clock ..... 332
7.3.15 SMBus Clocks ..... 332
7.3.16 Debug and DFx Clocks ..... 332
7.3.17 Clocks Generated During the S5 Sleep State ..... 332

**8 Interrupts and Events ..... 333**
8.1 PCI Interrupts and Routing ..... 333
8.1.1 Interrupt Pin Route Overview ..... 335
8.1.2 PIRQx Register Decode Table ..... 337
8.1.3 Edge- and Level-Triggered Mode ..... 337
8.2 Non-Maskable Interrupt (NMI) ..... 338
8.3 System Management Interrupt (SMI) ..... 338
8.4 System Control Interrupt (SCI) ..... 339
8.5 GPIO and PMC General Purpose Event (GPE) ..... 340
8.6 Message Signaled Interrupt (MSI and MSI-X) ..... 343
8.7 I/O APIC Input Mapping ..... 343
8.8 8259 PIC Input Mapping ..... 345
8.9 Device Interrupt-Generating Capabilities ..... 346

**9 Memory Controller ..... 347**
9.1 Introduction ..... 347
9.2 Signal Descriptions ..... 348
9.3 Feature List ..... 353
9.4 Memory Controller Power Management Features ..... 354
9.4.1 DRAM Running Average Power Limit (RAPL) ..... 354
9.4.2 Dynamic (HW autonomous) Self-refresh DDR4 ..... 354
9.4.3 Unsupported Features and Restrictions ..... 355
9.4.4 DDR4 DIMM Types and Population ..... 356
9.4.4.1 DIMM Population Requirements ..... 357
9.5 Memory Controller RAS Features ..... 358
9.5.1 Data Scrambler ..... 358
9.5.2 Patrol Scrub ..... 358
9.5.3 Demand Scrub ..... 358
9.5.4 Asynchronous DRAM Refresh (ADR) ..... 358

**10 Flexible I/O Adapter (FIA) Overview....359**
10.1 Signal Descriptions....360
10.2 Feature List....360
10.3 Architectural Overview....361
10.4 USB 3.0 as a Debug Port....363
10.5 Lane Configuration....363
10.5.1 Intel® Management Engine and BIOS Roles in Lane Configuration....363
10.5.2 PCI Express Lane Configuration....365
10.5.3 Configuring PCIe Lane to x1 and SATA Ports....366
10.6 Register Map....367
10.6.1 PCI Configuration and Capabilities....368
10.6.2 Sideband Registers....368
**11 PCI Express Root Ports (RP)....369**
11.1 Signal Descriptions....371
11.2 Feature List....372
11.3 Architectural Overview....374
11.3.1 PCI Bridge Subsystem Identification Capability....375
11.3.2 Message Signaled Interrupt (MSI) Capability....375
11.3.3 Advanced Error Reporting (AER) Capability....375
11.3.4 PCIe Extended Capability and Other Supported Technology....376
11.3.4.1 Access Control Services (ACS) Capability....376
11.3.4.2 Windows Hardware Error Architecture (WHEA)....376
11.3.4.3 Dynamic Link Width....377
11.3.4.4 Alternative Routing-ID Interpretation (ARI)....377
11.4 Physical Layer....378
11.4.1 PCI Express Speed Support....378
11.4.2 Separate Refclk with Independent SSC Architecture (SRIS)....378
11.4.3 Form Factor Support....378
11.4.4 Configuration of the PCI Express Ports and Link Widths....379
11.4.5 Degraded Mode....380
11.4.5.1 Dynamic Lane Reversal....381
11.4.5.2 Lane Polarity Inversion....381
11.5 Transaction Layer....382
11.5.1 Transaction ID....382
11.5.2 Attributes....383
11.5.3 Traffic Class....384
11.5.4 Completer ID....384
11.5.5 Transaction Ordering....385
11.5.6 PCI Express Port Arbitration....385
11.5.7 Request Splitting....385
11.5.8 Read Prefetching Policies....386
11.5.9 Read Completion Combining....386
11.5.10 Write Combining....386
11.6 Interrupt Support....387
11.6.1 Legacy Interrupt Sharing....387
11.7 Power Management....388
11.7.1 Hardware Controlled Active Power Management....388
11.7.2 PCI-PM 1.2 Support....388
11.7.3 Power Management Event Signaling....389
11.7.4 Beacon and WAKE# Signaling....389
11.7.5 Non-D0 State PM Handling....390
11.7.5.1 Non-D0 Behavior for Transmit (Tx) Transactions....390
11.7.5.2 Non-D0 Behavior for Receive (Rx) Transactions....390
11.7.6 BME Clear State Handling....391

11.7.6.1 BME (=0) Clear State Behavior for Transmit (Tx) Transactions ..... 391
11.7.6.2 BME (=0) Clear State Behavior for Receive (Rx) Transactions ..... 391
11.7.6.3 Internal MSI/Error Messages During Non-D0/BME (clear) ..... 391
11.7.6.4 L1 PM Substates ..... 391
11.8 PCI Express RAS Features ..... 392
11.8.1 End-to-End CRC (ECRC) Support ..... 392
11.8.2 Completion Timeout ..... 392
11.8.3 Data Poisoning ..... 392
11.8.4 Role-based Error Reporting ..... 393
11.9 PCI Express Hot-Plug Surprise ..... 394
11.9.1 PCI Express Hot-Plug Interrupts and Events ..... 394
11.9.1.1 Presence Detect ..... 394
11.9.1.2 SCI/SMI Generation ..... 394
11.10 Register Map ..... 395

**12 SATA Controller ..... 396**
12.1 Signal Descriptions ..... 397
12.2 Feature List ..... 398
12.3 Architectural Overview ..... 398
12.4 AHCI Operation ..... 399
12.4.1 Staggered Spin Up Support ..... 399
12.4.2 Staggered Spin-up Operation in AHCI Mode ..... 399
12.4.3 Interrupts ..... 400
12.4.3.1 Interaction Between INTx# and MSI/MSI-X ..... 400
12.4.4 Power Management ..... 400
12.4.5 Automatic Transition of Partial to Slumber ..... 400
12.4.6 Enclosure Management and Activity LED ..... 400
12.5 Register Map ..... 401
12.5.1 PCI Configuration and Capabilities ..... 402
12.5.2 Memory-Mapped Registers ..... 404
12.5.3 I/O-Mapped Registers ..... 409

**13 LAN Controllers ..... 410**
13.1 Signal Descriptions ..... 412
13.2 Supported Modes of Operation ..... 415
13.3 Feature List ..... 417
13.4 Programmers Reference Manual ..... 421
13.5 Architectural Overview ..... 421
13.5.1 PCIe Integrated Endpoint ..... 422
13.5.2 Interconnect Interfaces ..... 423
13.6 Basic Operation ..... 424
13.6.1 Transmit (Tx) Data Flow ..... 424
13.6.2 Receive (Rx) Data Flow ..... 425
13.7 Interconnects ..... 426
13.7.1 Setting Up PCI Device Presence and Non-presence ..... 426
13.7.1.1 LAN Controllers Soft Straps ..... 426
13.7.2 Disabling PCI Functions by BIOS ..... 426
13.7.3 LAN Port Interface ..... 427
13.7.4 Reference Clock Input ..... 428
13.7.5 Pin Straps ..... 428
13.8 Shared SPI Flash for Use with LAN0 and LAN1 ..... 429
13.8.1 Shared SPI Flash Starter Images ..... 429
13.8.2 Shared SPI Flash Map Related to LAN0 and LAN1 ..... 429
13.8.3 Unique MAC Address ..... 429

13.8.4 VLAN Support ..... 430
13.9 MMIO and Software Interface ..... 430
13.10 Host Primary Interface ..... 431
13.10.1 Host Interface Architecture, Transaction and Link Layer Properties ..... 431
13.10.2 PCIe Transaction Layer ..... 432
13.10.2.1 Transaction Types Accepted by the LAN Controllers ..... 432
13.10.2.2 Transaction Types Initiated by the LAN Controllers ..... 434
13.10.2.3 Messages ..... 436
13.10.2.4 Transaction Attributes ..... 436
13.10.2.5 Ordering Rules ..... 437
13.10.3 Error Events and Error Reporting ..... 439
13.10.3.1 General Description ..... 439
13.10.3.2 Error Events ..... 440
13.10.3.3 Completion Timeout Mechanism ..... 441
13.10.3.4 Error Forwarding (TLP Poisoning) ..... 441
13.10.3.5 Completion With Unsuccessful Completion Status ..... 441
13.10.3.6 Blocking on Upper Address ..... 442
13.10.3.7 Proprietary Error Reporting ..... 442
13.11 Management Interfaces ..... 444
13.11.1 SMBus ..... 444
13.11.1.1 Channel Behavior ..... 444
13.11.2 NC-SI Transactions ..... 444
13.11.3 MCTP (over SMBus) ..... 444
13.12 Sideband Interface (NC-SI) ..... 445
13.12.1 Electrical Characteristics ..... 445
13.13 Shared SPI Flash ..... 445
13.13.1 General Overview ..... 445
13.13.2 Shared SPI Flash Protection ..... 446
13.13.3 Shadow RAM ..... 447
13.13.3.1 Shadow RAM Update Flow ..... 448
13.13.4 Shared SPI Flash Clients and Interfaces ..... 449
13.13.5 Memory Mapped Host Interface ..... 450
13.13.6 Shared SPI Flash Access Contention ..... 451
13.13.6.1 Flash Deadlock Avoidance ..... 452
13.13.7 Signature Field ..... 453
13.13.8 VPD Support ..... 453
13.13.8.1 VPD Access Flows ..... 455
13.13.8.2 VPD Area Update Flows ..... 455
13.13.8.3 Shared SPI Flash Update Read, Write, and Erase Sequences ..... 455
13.13.8.4 Software Flow to the Bit Banging Interface ..... 455
13.13.8.5 Erase Flow Using the FLA Register ..... 456
13.13.8.6 Software Access flow to Shadow RAM ..... 456
13.13.8.7 Flash Access Flow via the Memory Mapped Interface ..... 457
13.13.8.8 Software Flash/Shadow RAM Program Flow ..... 457
13.13.8.9 Software Flash Read Flow via the Flash-mode Interface ..... 457
13.13.9 Extended Shared SPI Flash Update Flows ..... 458
13.13.9.1 Flow for Updating Secured Modules ..... 458
13.14 Configurable I/O Pins — Software-Definable Pins (SDPs) ..... 460
13.15 LEDs ..... 462
13.16 Network Management Interface (MDIO or I2C) ..... 464
13.16.1 $I^2C$ or MDIO Selection ..... 464
13.16.2 Recommended Networking Topologies ..... 465
13.16.3 Management Data I/O Interface (MDIO) ..... 467
13.16.3.1 MDIO Timing Relationship to MDC ..... 469
13.16.3.2 IEEE802.3 Clause 22 and Clause 45 Differences ..... 470
13.16.3.3 MDIO Management Frame Structure ..... 470

13.16.3.4 MDIO Direct Access ..... 473
13.16.4 I2C ..... 474
13.16.4.1 Hardware Based I2C Access ..... 474
13.16.4.2 Bit Bang Based I2C Access ..... 475
13.16.4.3 Supported Commands ..... 475
13.17 Network Interface ..... 477
13.17.1 Companion Integrated PHY Register Access ..... 477
13.17.1.1 PHY Register Addressing ..... 477
13.17.1.2 PHY Indirect Read Operation ..... 478
13.17.1.3 PHY Indirect Write Operation ..... 478
13.17.2 Ethernet Flow Control (FC) ..... 479
13.17.2.1 MAC Control Frames and Reception of Flow Control Packets ..... 480
13.17.2.2 PAUSE and MAC Control Frames Forwarding ..... 484
13.17.2.3 Transmitting PAUSE Frames ..... 485
13.17.3 Inter Packet Gap (IPG) Control and Pacing ..... 490
13.18 Physical Functions (PF), Virtual Functions (VF) and ARI (Alternative RID Interpretation) ..... 491
13.19 Register Map ..... 492

**14 USB Combo Controller ..... 493**
14.1 Signal Descriptions ..... 494
14.2 Feature List ..... 495
14.2.1 USB 3.0 ..... 495
14.2.2 USB 2.0 ..... 495
14.2.3 xHCI Interface ..... 497
14.3 Architectural Overview ..... 498
14.3.1 xHCI Interface Architecture ..... 501
14.4 xHCI Data Structures ..... 503
14.5 Command Interface ..... 503
14.6 Root Hub Management ..... 504
14.7 Operational Model ..... 504
14.8 Supported Transaction Types ..... 505
14.9 Reset Requirements ..... 505
14.10 USB Power Management – Link States ..... 506
14.11 Error Condition and Handling ..... 507
14.12 Debug Capability (DbC) ..... 507
14.13 Register Map ..... 508
14.13.1 PCI Configuration and Capabilities ..... 509
14.13.2 Memory-Mapped Registers ..... 511

**15 Platform Control Cluster (PCC) ..... 522**
15.1 Signal Descriptions ..... 523
15.2 Architectural Overview ..... 523
15.2.1 Integrated PCI Devices ..... 523
15.2.2 Integrated PCI Device 31 Functions ..... 524
15.2.3 Legacy Controller Group ..... 524
15.2.3.1 Interrupt Timer Sub System (ITSS) ..... 524

**16 Power Management Controller (PMC) ..... 525**
16.1 Signal Descriptions ..... 526
16.2 Feature List ..... 528
16.3 Architectural Overview ..... 528
16.4 System Power States ..... 528
16.5 SoC Internal Power Control ..... 528
16.5.1 Internal Power Switching ..... 529
16.6 Total Cost of Ownership (TCO) ..... 530

16.6.1 Detecting a System Lockup ..... 531
16.6.2 Handling an Intruder ..... 531
16.6.3 SMM-to-OS and OS-to-SMM Calls ..... 532
16.7 SMI#/SCI Generation ..... 532
16.7.1 PCI Express* SCI ..... 534
16.8 C States ..... 534
16.9 Dynamic LPC 24-MHz Clock Control ..... 534
16.10 Sleep States ..... 535
16.10.1 Initiating State Changes when in the G0 (S0) Working State ..... 535
16.10.2 Initiating State Changes when in the G3 Mechanical-Off State ..... 537
16.10.3 Returning to the S0 State when in the G2 (S5) Soft-Off State ..... 537
16.10.3.1 LAN Power Management Wakeup ..... 539
16.10.3.2 Other Internal Power Management Events ..... 540
16.10.4 Reset Triggers That Generate a State Change ..... 541
16.11 System Power Supplies, Planes, and Signals ..... 542
16.11.1 Platform Power Plane Control with PMU_SLP_S3_N and PMU_SLP_S45_N ..... 542
16.11.2 COREPWROK SoC Input Signal ..... 542
16.11.3 SUSPWRDNACK SoC Output Signal ..... 542
16.12 Legacy Power Management Theory of Operation ..... 543
16.13 Reset Behavior ..... 543
16.13.1 Reset Request Conditions ..... 544
16.14 Interaction with the Innovative Engine (IE) ..... 546
16.15 Asynchronous DRAM Refresh (ADR) ..... 547
16.15.1 Configuring ADR ..... 548
16.15.1.1 Qualified Internal Events ..... 549
16.15.2 Configuring the Copy-to-Flash Extension ..... 550
16.15.3 NVDIMM Copy-to-Flash ..... 550
16.15.4 ADR Sequence of Events ..... 551
16.15.5 BIOS after Successful ADR ..... 552
16.16 Register Map ..... 553
16.16.1 PCI Configuration and Capabilities ..... 554
16.16.2 Memory-Mapped Registers ..... 554
16.16.3 I/O-Mapped Registers ..... 555
16.16.4 Sideband Registers ..... 556

**17 Embedded Multi-Media Card (eMMC) ..... 557**
17.1 Signal Descriptions ..... 558
17.2 Feature List ..... 558
17.3 Architectural Overview ..... 558
17.3.1 eMMC Mass Storage Command and Data Flows ..... 558
17.3.2 PCI Mode and ACPI Mode ..... 559
17.4 Address Map ..... 559
17.5 Register Map ..... 560
17.5.1 PCI Configuration and Capabilities ..... 561
17.5.2 Memory-Mapped Registers ..... 562
17.5.3 I/O-Mapped Registers ..... 564

**18 High-Speed UART Controller ..... 565**
18.1 Signal Descriptions ..... 566
18.2 Feature List ..... 568
18.2.1 UART Features ..... 568
18.2.2 DMA Features ..... 568
18.3 Architectural Overview ..... 569
18.3.1 UART Hiding and Locking ..... 570
18.3.2 Legacy Mode ..... 571

18.3.3 Enhanced Mode ..... 571
18.3.4 UART Function ..... 572
18.3.5 FIFO Interrupt Mode Operation ..... 573
18.3.5.1 Receive Interrupt ..... 573
18.3.5.2 Timeout Interrupt ..... 573
18.3.5.3 Transmit Interrupt ..... 573
18.3.6 FIFO Polled Mode Operation ..... 574
18.3.7 Auto-flow Control Operation ..... 574
18.3.8 Loopback Operation ..... 575
18.3.9 Auto-Baud Detection ..... 575
18.3.10 UART Clocking ..... 576
18.3.11 UART Baud Rate Generation ..... 576
18.3.12 I/O and Memory Base Access Space ..... 577
18.3.13 Interrupt ..... 577
18.3.14 Memory Address MAP ..... 577
18.3.15 DMA ..... 578
18.3.15.1 Channel Arbitration ..... 579
18.3.15.2 Descriptor Arbitration ..... 580
18.3.16 MSI Generation ..... 581
18.4 UART Programming Recommendation ..... 582
18.4.1 Setup Access Space ..... 582
18.4.2 Setup Interrupt ..... 582
18.4.3 Space Enable ..... 582
18.4.4 DMA Programming ..... 582
18.4.5 DMA Descriptor Use/Reuse ..... 583
18.4.6 Programming Optimization ..... 583
18.4.6.1 Optimization Descriptor Programming Activities During DMA ..... 583
18.4.6.2 Optimization Access Time By UART During DMA ..... 583
18.4.6.3 Optimization Interrupt Handling ..... 584
18.5 Register Map ..... 587
18.5.1 PCI Configuration and Capabilities Registers ..... 588
18.5.2 Memory Mapped Registers, BAR1 (Memory Base Address Register, MEMBA) ..... 589
18.5.2.1 UART Memory Mapped Registers ..... 589
18.5.3 DMA Channel Registers ..... 590
18.5.4 I/O Mapped Registers, BAR 0 (I/O Base Address Register, IOBA) ..... 591
18.5.4.1 UART I/O Mapped Registers ..... 591

**19 Serial Peripheral Interface (SPI) ..... 592**
19.1 Signal Descriptions ..... 593
19.2 Feature List ..... 593
19.3 Architectural Overview ..... 594
19.3.1 Operation Mode ..... 595
19.3.1.1 Descriptor Mode ..... 595
19.3.1.2 Security ..... 596
19.3.1.3 SPI Flash Access ..... 596
19.3.1.4 Flash Component Requirements ..... 597
19.3.2 Flash Regions ..... 598
19.3.2.1 Flash Component Boundary ..... 600
19.3.2.2 Flash Descriptor Region 0 ..... 600
19.4 Flash Access ..... 602
19.4.1 Direct Access ..... 602
19.4.1.1 BIOS Direct Read in Descriptor Mode ..... 603
19.4.1.2 Chip Select and Flash Linear Address (FLA) ..... 604
19.4.2 Programmed Register Access ..... 604
19.5 Flash Security and Protection ..... 605

19.5.1 Flash Range Write and Read Protection ..... 605
19.5.2 SMI Based Global Write Protection ..... 606
19.5.3 Flash Security Override Strap ..... 606
19.5.4 Illegal Instructions ..... 606
19.5.5 Error Status Table ..... 607
19.5.6 Region Checks ..... 607
19.6 SMI Generation ..... 608
19.6.1 Synchronous SMI ..... 608
19.6.2 Asynchronous SMI ..... 608
19.7 Hardware vs. Software Sequencing ..... 608
19.7.1 Hardware Sequencing ..... 609
19.7.2 Software Sequencing ..... 610
19.8 Serial Flash Device Compatibility Requirements ..... 612
19.8.1 BIOS SPI Flash Requirements ..... 612
19.9 SPI Flash Interface ..... 613
19.9.1 Single-Input, Dual-Output Fast Read ..... 613
19.9.2 Dual-I/O Fast Read ..... 614
19.9.3 Quad-Output Fast Read ..... 615
19.9.4 Quad-I/O Fast Read ..... 616
19.9.5 JEDEC ID ..... 616
19.10 SPI Support for Trusted Platform Module (TPM) ..... 617
19.10.1 TPM Protocol on SPI ..... 618
19.10.2 TPM Flow Control on SPI ..... 620
19.11 Register Map ..... 622
19.11.1 PCI Configuration and Capabilities ..... 623
19.11.2 Memory Mapped Registers ..... 624

**20 Low Pin Count (LPC) Controller ..... 626**
20.1 Signal Descriptions ..... 627
20.2 Feature List ..... 628
20.3 Architectural Overview ..... 628
20.3.1 LPC Cycle Types ..... 628
20.3.2 Memory Cycle ..... 628
20.3.3 I/O Cycle ..... 628
20.3.4 DMA Cycle ..... 628
20.3.5 Bus Master Cycle Notes ..... 628
20.3.6 LFRAME# Usage ..... 628
20.3.7 LPC PD# Protocol ..... 629
20.3.8 LPC Interface Decoders ..... 629
20.3.9 SERR# Generation ..... 629
20.3.10 Dynamic LPC Clock Control ..... 629
20.3.10.1 Verifying if Safe to Stop the LPC Clock ..... 629
20.3.10.2 Conditions for Maintaining the LPC Clock ..... 629
20.3.10.3 Conditions for Stopping the LPC Clock ..... 630
20.3.10.4 Conditions for Restarting the LPC Clock ..... 630
20.4 Register Map ..... 631
20.4.1 PCI Configuration and Capabilities ..... 632
20.4.2 Fixed I/O Address Transactions ..... 633
20.4.3 Configurable I/O Address Transactions ..... 635
20.4.4 Memory and BIOS Address Transactions ..... 636
20.4.4.1 BIOS Write Protect and Top Swap Range ..... 637
20.4.4.2 BIOS Boot Decode Claiming Agent ..... 637

**21 PIC and I/O APIC Controllers ..... 638**
21.1 Signal Descriptions ..... 639
21.2 Programmable Interrupt Controller (PIC) ..... 639

21.2.1 Special Internal Circuitry ..... 639
21.2.2 Generating Interrupts ..... 639
21.2.3 Acknowledging Interrupts ..... 640
21.2.4 Hardware/Software Interrupt Sequence ..... 640
21.2.5 Initialization Command Words (ICWx) ..... 641
21.2.5.1 ICW1 ..... 641
21.2.5.2 ICW2 ..... 641
21.2.5.3 ICW3 ..... 641
21.2.5.4 ICW4 ..... 641
21.2.6 Operation Command Words (OCW) ..... 642
21.2.7 Fully-Nested Mode ..... 642
21.2.8 Special Fully-Nested Mode ..... 642
21.2.9 Automatic Rotation Mode (Equal Priority Devices) ..... 642
21.2.10 Specific Rotation Mode (Specific Priority) ..... 643
21.2.11 Poll Mode ..... 643
21.2.12 Edge- and Level-Triggered Mode ..... 643
21.2.13 End of Interrupt (EOI) Operations ..... 644
21.2.13.1 Normal End of Interrupt (EOI) Mode ..... 644
21.2.13.2 Automatic End of Interrupt (AEOI) Mode ..... 644
21.2.14 Masking Interrupts ..... 645
21.2.14.1 Masking on an Individual Interrupt Request ..... 645
21.2.14.2 Special Mask Mode ..... 645
21.3 I/O Advanced Programmable Interrupt Controller (IOxAPIC) ..... 646
21.3.1 Interrupt Redirection Table ..... 646
21.3.2 Accessing the Redirection Table ..... 646
21.3.3 IOxAPIC Interrupts to the Local APICs ..... 647
21.3.4 Interrupt Delivery Modes ..... 649
21.3.5 Determining Safe Message Delivery Time ..... 649
21.3.6 End Of Interrupt (EOI) Register ..... 649
21.4 Register Map ..... 650
21.4.1 PCI Configuration and Capabilities ..... 651
21.4.2 Memory-Mapped Registers ..... 652
21.4.3 I/O-Mapped Registers ..... 653

**22 Primary to Side Band (P2SB) Bridge ..... 654**
22.1 Signal Descriptions ..... 655
22.2 Feature List ..... 655
22.3 Architectural Overview ..... 655
22.4 Register Map ..... 656
22.4.1 PCI Configuration and Capabilities ..... 657
22.4.2 Memory-Mapped Registers ..... 658

**23 Programmable Interval Timer (PIT) ..... 659**
23.1 Signal Descriptions ..... 660
23.2 Feature List ..... 660
23.3 Architectural Overview ..... 660
23.3.1 Counter 0, System Timer ..... 661
23.3.2 Counter 2 ..... 661
23.4 Counter Mode Definitions ..... 662
23.5 Timer Programming ..... 662
23.5.1 Writing to the Interval Timer ..... 662
23.5.2 Reading from the Interval Timer ..... 663
23.5.2.1 Simple Read ..... 663
23.5.2.2 Counter Latch Command ..... 663
23.5.2.3 Read Back Command ..... 664
23.6 Register Map ..... 665

23.6.1 I/O-Mapped Registers ..... 666

**24 Real Time Clock (RTC) ..... 667**
24.1 Signal Descriptions ..... 668
24.2 Feature List ..... 669
24.3 Architectural Overview ..... 669
24.4 I/O Registers ..... 670
24.4.1 Alternate-Access Mode ..... 671
24.5 Update Cycles ..... 672
24.6 Interrupts ..... 672
24.7 Lockable RAM Ranges ..... 672
24.8 Month and Year Alarms ..... 672
24.9 Register Map ..... 673
24.9.1 PCI Configuration and Capabilities ..... 674
24.9.2 Memory-Mapped Registers ..... 674
24.9.3 I/O-Mapped Registers ..... 674

**25 High Precision Event Timer (HPET) ..... 675**
25.1 Signal Descriptions ..... 675
25.2 Feature List ..... 675
25.3 Architectural Overview ..... 676
25.3.1 Timer Accuracy ..... 677
25.3.2 Timer Off-Load ..... 677
25.3.3 Interrupt Mapping ..... 678
25.3.3.1 Mapping Option #1 (Legacy Replacement Option) ..... 678
25.3.3.2 Mapping Option #2 (Standard Option) ..... 679
25.3.3.3 Mapping Option #3 (Processor Message Option) ..... 679
25.3.4 Periodic Versus Non-Periodic Modes ..... 680
25.3.4.1 Non-Periodic (One-Shot) Mode ..... 680
25.3.4.2 Periodic Mode ..... 680
25.3.5 Enabling the Timers ..... 681
25.3.6 Interrupt Levels ..... 681
25.3.7 Handling Interrupts ..... 681
25.3.8 Issues Related to 64-Bit Timers with 32-Bit Processors ..... 681
25.4 Register Map ..... 682
25.4.1 Memory-Mapped Registers ..... 683

**26 Customer General-Purpose I/O (GPIO) ..... 685**
26.1 GPIO Ball and Signal Names ..... 686
26.2 Pad Description ..... 691
26.3 Signal Configuration Overview ..... 692
26.3.1 Pad Ownership ..... 693
26.3.2 Pad Multiplex Function ..... 694
26.3.3 Pad Internal Pull-up / Pull-down Configure ..... 694
26.3.4 Pad Configuration Logical Control Summary ..... 695
26.3.5 Non-Maskable Interrupt (NMI) ..... 696
26.3.6 System Management Interrupt (SMI) ..... 696
26.3.7 System Control Interrupt (SCI)/General Purpose Event (GPE) ..... 696
26.3.8 GPIO to IOxAPIC ..... 697
26.3.9 Legacy Interrupt Request (IRQ) ..... 697
26.3.10 Pad Level/Edge Sensitive Detect Enable ..... 697
26.4 Registers Lock Down ..... 698
26.5 MMIO Address Access ..... 698

**27 System Management Bus (SMBus) Controller - Overview ..... 706**
27.1 General Architecture ..... 707
27.2 Feature List ..... 709

**28 SMBus Controller - Legacy ..... 711**
28.1 Signal Descriptions ..... 712
28.2 General Architecture ..... 713
28.3 Legacy SMBus Controller Operation ..... 714
28.3.1 Supported SMBus Protocols ..... 714
28.3.2 SMBus CRC Generation and Checking ..... 715
28.4 Protocols as an SMBus Master ..... 716
28.4.1 Quick Command ..... 716
28.4.2 Send Byte and Receive Byte Commands ..... 716
28.4.3 Write Byte and Write Word Commands ..... 716
28.4.4 Read Byte and Read Word Commands ..... 717
28.4.5 Process Call Command ..... 717
28.4.6 Block Read and Block Write Commands ..... 718
28.4.7 Block Write-Block Read Process Call Command ..... 719
28.4.8 SMBus Host Notify Command ..... 719
28.5 Protocols and Commands as the ARP Master ..... 720
28.5.1 ARP Enumeration and the Notify ARP Master Command ..... 720
28.6 $I^2C$ Read Command ..... 721
28.7 Bus Arbitration as an SMBus Master ..... 722
28.7.1 SMBCLK Signal Stretching by an SMBus Slave ..... 722
28.7.2 SMBus Timeout ..... 722
28.8 Interrupts and SMI ..... 722
28.9 Optional SMBALERT# Signal ..... 724
28.10 Register Map ..... 725
28.10.1 Registers in the Configuration Space ..... 726
28.10.2 Registers in the Memory Space ..... 727
28.10.3 Registers in the I/O Space ..... 728

**29 SMBus Controller - Host ..... 729**
29.1 Signal Descriptions ..... 730
29.2 Features ..... 730
29.3 Architectural Overview ..... 731
29.4 Controller Characteristics and Operation ..... 733
29.4.1 SMBus Behavior on PCIe Reset ..... 733
29.4.2 Addressing and Configuration ..... 733
29.4.2.1 ARP Nomenclature ..... 734
29.4.2.2 Unique Device Identifier (UDID) Format ..... 735
29.4.2.3 ARP Slave Behavior ..... 736
29.4.2.4 ARP Master Behavior ..... 743
29.4.2.5 ARP Initialization Flow ..... 746
29.4.3 SMT System Usage Models ..... 748
29.4.4 SMT Security Requirements ..... 748
29.4.5 SMT Timing Modes ..... 748
29.4.6 SMT as Master ..... 749
29.4.6.1 Hardware Buffering for Master Support ..... 749
29.4.6.2 Master Descriptor ..... 750
29.4.6.3 Master Descriptor Usage ..... 754
29.4.6.4 Master Transactions Flow ..... 758
29.4.6.5 Clearing of the Start Bit ..... 761
29.4.6.6 Master Retry Flow ..... 762
29.4.6.7 Write Disabling to DIMM SPD EEPROM Addresses ..... 764
29.4.7 SMT as Target ..... 765
29.4.7.1 Hardware Buffering for Target Support ..... 765
29.4.7.2 Target Descriptor ..... 766
29.4.7.3 Target Transaction Status ..... 770
29.4.7.4 Target Memory Buffer Hardware-Firmware Flow ..... 772

29.4.7.5 Target Flow ..... 775
29.4.8 Dynamic SMT Policy Update ..... 783
29.4.8.1 Master Policy ..... 783
29.4.8.2 Target Policy ..... 783
29.5 Interrupts ..... 785
29.5.1 Master Interrupts ..... 786
29.5.2 Target Interrupts ..... 787
29.5.3 Error Interrupts ..... 788
29.5.4 Interrupt Cause Logging ..... 789
29.6 SMT RAS Architecture ..... 790
29.6.1 Soft Reset (DEVCTL.IFLR and GCTRL.SRST) ..... 790
29.6.2 Target Reset (GCTRL.TRST) ..... 791
29.7 MCTP Over SMBus Packet Header Format ..... 792
29.8 Register Map ..... 794

**30 SMBus Controller - Platform Environment Control Interface ..... 795**

30.1 Signal Descriptions ..... 796
30.2 PECI over SMBus Features ..... 796
30.3 SMBus Supported Transactions ..... 797
30.4 SMBus Block Read/Write Transaction Formats ..... 798
30.5 SMBus Commands ..... 798
30.6 PECI Over SMBus ..... 799
30.6.1 PECI Message Header in SMBus ..... 799
30.6.1.1 Target Address Field ..... 799
30.6.1.2 Write Length Field ..... 799
30.6.1.3 Read Length Field ..... 799
30.6.1.4 Command Byte ..... 799
30.6.2 PECI Write-Read Protocol ..... 800
30.6.2.1 PECI Proxy Command Format ..... 801
30.6.2.2 PECI Proxy Read Command ..... 802
30.6.3 PECI Proxy Command Handling Procedure ..... 804
30.6.4 PECI Proxy Command Trigger ..... 805
30.6.4.1 Unsupported PECI Command ..... 805
30.6.4.2 Illegally Formatted Command ..... 805
30.7 PECI Proxy Commands ..... 806
30.7.1 Ping() ..... 807
30.7.2 GetDIB() ..... 809
30.7.2.1 PECI Device Info Field ..... 811
30.7.2.2 PECI Revision Number ..... 811
30.7.3 GetTemp() ..... 812
30.7.4 RdPkgConfig() ..... 814
30.7.5 WrPkgConfig() ..... 816
30.7.6 RdIAMSR() ..... 818
30.7.6.1 Processor ID Enumeration ..... 818
30.7.7 RdPCIConfigLocal() ..... 821
30.7.8 WrPCILocalConfig() ..... 824
30.7.9 RdEndPointConfig() ..... 827
30.7.10 WrEndPointConfig() ..... 829
30.7.11 MCA Access via RdEndPointConfig() ..... 831
30.7.12 Endpoints supporting Rd/WrPCILocalConfig ..... 832
30.7.13 Endpoints supporting Rd/WrEndPointConfig ..... 833
30.8 DRAM Thermal Capabilities ..... 834
30.8.1 DRAM Rank Temperature Write (Index = 18) ..... 836
30.8.2 DIMM Temperature Read (Index = 14) ..... 836
30.8.3 DRAM Channel Temperature Read (Index = 22) ..... 837
30.8.4 DRAM Accumulated Energy Read (Index = 4) ..... 837

30.8.5 DRAM Power Info Read (Index = 35/36) ..... 838
30.8.6 DRAM Power Limit Data Read/Write (Index = 34) ..... 839
30.8.7 DRAM Power Limit Performance Status Read ..... 840
30.9 CPU Thermal and Power Optimization Capabilities ..... 841
30.9.1 Package Identifier Read (Index = 0) ..... 848
30.9.1.1 CPU ID Information ..... 848
30.9.1.2 Platform ID ..... 848
30.9.1.3 PCU Device ID ..... 849
30.9.1.4 Maximum Thread ID ..... 849
30.9.1.5 CPU Microcode Update Revision ..... 849
30.9.1.6 MCA Error Source Log ..... 850
30.9.2 Package Power SKU Unit Read (Index = 30) ..... 850
30.9.3 Package Power SKU Read (Index = 28 and 29) ..... 851
30.9.4 Accumulated Run Time Read (Index = 31) ..... 852
30.9.5 Package Temperature Read (Index = 2) ..... 852
30.9.6 Per Core DTS Temperature Read (Index = 9) ..... 852
30.9.7 Temperature Target Read (Index = 16) ..... 853
30.9.8 Package Thermal Status Read / Clear (Index = 20) ..... 853
30.9.9 Thermally Constrained Time Read (Index = 32) ..... 854
30.9.10 Current Limit Read (Index = 17) ..... 854
30.9.11 Accumulated Energy Status Read (Index = 3) ..... 854
30.9.12 Package Power Limits For Multiple Turbo Modes (Index = 26 and 27) ..... 855
30.9.13 Package Power Limit Performance Status Read (Index = 8) ..... 856
30.9.14 Efficient Performance Indicator Read (Index = 6) ..... 856
30.9.15 ACPI P-T Notify Write and Read (Index = 33) ..... 857
30.9.16 Package PL3 (Index = 53) ..... 858
30.9.17 Read Ratio Constraints (Index = 49) ..... 859
30.9.18 Read Ratio Limit ..... 860
30.9.19 Write Ratio Limit ..... 860
30.10 Crash Dump ..... 861
30.10.1 GetCrashDumpCapabilities (Index=51) ..... 861
30.10.2 GetCrashDumpFrame (Index = 52) ..... 862
30.10.3 UnCore and PM Crash Dump Registers ..... 863
30.10.4 Core Crash Dump ..... 866
30.10.4.1 Overview ..... 866
30.10.4.2 PECI Command Structure ..... 866
30.10.4.3 Flow ..... 868
30.10.4.4 Core Crash Details ..... 869
30.11 DTS Temperature Data ..... 871
30.11.1 PECI Device Temp Data ..... 871
30.11.2 Interpretation ..... 871
30.11.3 Temperature Filtering ..... 872
30.11.4 Reserved Values ..... 872

**31 Intel® QuickAssist Technology (Intel® QAT) ..... 873**

31.1 Signal Descriptions ..... 874
31.2 Features ..... 874
31.3 Usage Model ..... 875
31.3.1 Ring Controller ..... 876
31.3.2 Single-Root I/O Virtualization (SR-IOV) ..... 876
31.3.2.1 I/O Virtualization Models ..... 876
31.3.2.2 Sharing via Virtual Machine Monitor (VMM) ..... 876
31.3.2.3 Direct Assignment of EP with SR-IOV ..... 877
31.3.3 Ring Bundle Mapping ..... 878
31.3.3.1 VF Interrupts ..... 878

31.3.4 PCIe Endpoint (EP) Function ..... 879
31.3.4.1 Transmit Interface (TI) ..... 879
31.3.4.2 Receive Interface (RI) ..... 879
31.3.4.3 PCIe EP Functions ..... 880
31.3.4.4 EP Function Mapping ..... 880
31.3.4.5 EP Mapping of BARs to MMIO ..... 881
31.3.4.6 Alternative Routing-ID Interpretation (ARI) ..... 881
31.3.5 Interrupts ..... 882
31.3.5.1 INTx ..... 882
31.3.5.2 Message Signaled Interrupt (MSI) ..... 882
31.3.5.3 MSI-X ..... 882
31.3.6 EP Errors ..... 883
31.3.6.1 PCIe Error Management ..... 883
31.3.6.2 PCIe Error Reporting Mechanisms ..... 883
31.3.6.3 PCIe Error Handling and Signaling ..... 884
31.3.6.4 PCIe Error Sources ..... 884
31.3.7 Device-Specific Error Management ..... 889
31.3.7.1 Memory Error Poisoning ..... 889
31.3.8 Error Handling with SR-IOV ..... 889
31.4 PCI Configuration Space Map ..... 890

**32 SoC Voltage Rails ..... 891**
32.1 SoC Voltage (And Related Signals) Descriptions ..... 892

**33 Resets and Voltage Sequencing ..... 895**
33.1 Signal Descriptions ..... 895
33.2 Architectural Overview ..... 897
33.3 Cold Boot: From G3 To G0.S0 ..... 898
33.4 Shutdown: From G0.S0 To G3 ..... 900
33.5 Deep Sleep: From G0.S0 To G1.S4 ..... 902
33.6 Deep Sleep Wake: From G1.S4 To G0.S0 ..... 902
33.7 Soft Off: From G0.S0 To G2.S5 ..... 902
33.8 Soft Off Wake: From G2.S5 To G0.S0 ..... 902
33.9 Power Failure: From Gx.Sx To G3 ..... 902
33.10 Special Considerations for Platform Voltages ..... 902

**34 Power Management ..... 903**
34.1 Signal Descriptions ..... 903
34.2 Overview ..... 904
34.3 Power Management Features ..... 905
34.3.1 Asynchronous DRAM Refresh (ADR) ..... 905
34.4 Power Wells ..... 906
34.4.1 Core Power Well ..... 906
34.4.2 SUS Power Well ..... 906
34.4.3 RTC Power Well ..... 906
34.5 Supply Voltage Rails ..... 907
34.6 Serial Voltage Identification (sVID) Controller ..... 908
34.6.1 SVID VR Requirements ..... 908
34.6.1.1 sVID Commands ..... 909
34.7 Core Power Management Overview ..... 910
34.8 System Global Power States ..... 911
34.8.1 Idle Power Optimizations ..... 912
34.9 Processor Power States - C-States ..... 913
34.10 Performance States ..... 915
34.10.1 Processor Performance States - P-States ..... 915
34.10.1.1 Frequency/Voltage Scaling ..... 915
34.10.2 Software P-State Requests ..... 915

34.11 Power Management Technologies ..... 916
34.11.1 Intel® Turbo Boost Technology ..... 916
34.11.2 Voltage Regulator Constraints ..... 917
34.11.3 Thermal Design Power Constraints ..... 917
34.11.4 FAST PROCHOT ..... 917
34.11.5 Running Average Power Limiting (RAPL) ..... 918
34.11.6 Always-On Timers (AONT) ..... 918
34.11.7 I/O Device Controller Enable/Disable ..... 918

**35 Thermal Management ..... 919**
35.1 Signal Descriptions ..... 919
35.2 Feature Summary ..... 920
35.3 Registers with Multiple Access Methods ..... 921
35.4 Thermal Management (TM) ..... 921
35.4.1 Digital Thermal Sensor (DTS) ..... 922
35.4.2 Fan Speed Control (FSC) ..... 923
35.4.3 PROCHOT_N Signal Pin ..... 924
35.4.4 Thermal Throttling ..... 926
35.4.4.1 Throttle Threshold Temperature ..... 926
35.4.4.2 Core Frequency/Voltage Reduction (Intel® Thermal Monitor 2) ..... 927
35.4.4.3 Clock Modulation (Intel® Thermal Monitor 1) ..... 927
35.4.4.4 Software-Controlled Clock Modulation ..... 928
35.4.5 THERMTRIP_N ..... 928
35.4.6 Thermal Status and Interrupts ..... 929
35.4.6.1 Thermal Status and Interrupts — Processor ..... 929
35.4.6.2 Thermal Status and Interrupts — SoC Package ..... 932
35.5 Memory Thermal Management (MTM) ..... 934
35.5.1 Thermal Throttling ..... 934
35.5.1.1 Open Loop Thermal Throttling (OLTT) ..... 934
35.5.1.2 Close Loop Thermal Throttling (CLTT) ..... 935
35.5.1.3 CLTT Throttle Levels ..... 935
35.5.1.4 CLTT Temperature Threshold Values ..... 936
35.5.1.5 Legacy SMBus Used for Accessing Information for CLTT ..... 937
35.5.1.6 PECI SMBus Used for Accessing Information for CLTT ..... 937
35.5.2 MEM_THERMTRIP ..... 938
35.5.3 Thermal Status and Interrupts ..... 938
35.6 Run Average Power Limit (RAPL) ..... 939
35.6.1 RAPL — SoC Package ..... 940
35.6.2 RAPL — DRAM ..... 944
35.7 Inverse Temperature Dependence (ITD) ..... 945
35.8 Platform Voltage Regulator Thermal Alert ..... 945
35.9 Model-Specific Registers (MSR) ..... 946

**36 System Management ..... 947**
36.1 Signal Descriptions ..... 948
36.2 Intel® ME and Intel Server Platform Services Firmware ..... 949
36.2.1 Key Properties of Intel® ME ..... 949
36.2.2 Intel® ME Requirements ..... 949
36.3 Register Map ..... 950
36.3.1 Host Root Space - Intel® ME HECI1 ..... 951
36.3.1.1 PCI Configuration and Capabilities ..... 951
36.3.1.2 Memory-Mapped Registers ..... 953
36.3.1.3 I/O-Mapped Registers ..... 953
36.3.1.4 Sideband Registers ..... 953
36.3.2 Host Root Space - Intel® ME HECI2 ..... 954
36.3.2.1 PCI Configuration and Capabilities ..... 954

36.3.2.2 Memory-Mapped Registers ..... 955
36.3.2.3 I/O-Mapped Registers ..... 955
36.3.2.4 Sideband Registers ..... 955
36.3.3 Host Root Space - Intel® ME IDER ..... 956
36.3.3.1 PCI Configuration and Capabilities ..... 956
36.3.3.2 Memory-Mapped Registers ..... 956
36.3.3.3 I/O-Mapped Registers ..... 957
36.3.3.4 Sideband Registers ..... 958
36.3.4 Host Root Space - Intel® ME HECI3 ..... 959
36.3.4.1 PCI Configuration and Capabilities ..... 959
36.3.4.2 Memory-Mapped Registers ..... 960
36.3.4.3 I/O-Mapped Registers ..... 960
36.3.4.4 Sideband Registers ..... 960
36.3.5 Host Root Space - Intel® ME KT ..... 961
36.3.5.1 PCI Configuration and Capabilities ..... 961
36.3.5.2 Memory-Mapped Registers ..... 962
36.3.5.3 I/O-Mapped Registers ..... 963
36.3.5.4 Sideband Registers ..... 963

**37 Innovation Engine ..... 964**
37.1 Feature Summary ..... 966
37.2 Sample IE Usages ..... 967
37.3 Architectural Overview ..... 968
37.4 Inter-Processor Communication (IPC) ..... 971
37.4.1 IE Root Space Devices ..... 971
37.4.1.1 PECI ..... 971
37.4.1.2 SMBus Message Transport (SMT) ..... 971
37.4.1.3 UART ..... 971
37.5 Register Map ..... 972
37.5.1 Host Root Space - IE HECI1 ..... 973
37.5.1.1 PCI Configuration and Capabilities ..... 973
37.5.1.2 Memory-Mapped Registers ..... 975
37.5.1.3 I/O-Mapped Registers ..... 975
37.5.1.4 Sideband Registers ..... 975
37.5.2 Host Root Space - IE HECI2 ..... 976
37.5.2.1 PCI Configuration and Capabilities ..... 976
37.5.2.2 Memory-Mapped Registers ..... 977
37.5.2.3 I/O-Mapped Registers ..... 977
37.5.2.4 Sideband Registers ..... 977
37.5.3 Host Root Space - IE IDER ..... 978
37.5.3.1 PCI Configuration and Capabilities ..... 978
37.5.3.2 Memory-Mapped Registers ..... 979
37.5.3.3 I/O-Mapped Registers ..... 979
37.5.3.4 Sideband Registers ..... 980
37.5.4 Host Root Space - IE HECI3 ..... 981
37.5.4.1 PCI Configuration and Capabilities ..... 981
37.5.4.2 Memory-Mapped Registers ..... 982
37.5.4.3 I/O-Mapped Registers ..... 982
37.5.4.4 Sideband Registers ..... 982
37.5.5 Host Root Space - IE KT ..... 983
37.5.5.1 PCI Configuration and Capabilities ..... 983
37.5.5.2 Memory-Mapped Registers ..... 984
37.5.5.3 I/O-Mapped Registers ..... 984
37.5.5.4 Sideband Registers ..... 984

**38 JTAG and Debug Ports ..... 985**
38.1 Signal Descriptions ..... 986
38.2 Feature List ..... 987

38.3 Signals after Reset ..... 987
38.4 Intel® Direct Connect Interface (Intel® DCI) ..... 988
38.4.1 Debug Capability (DbC) ..... 988
38.4.2 Intel® DCI-OOB (Out of Band) ..... 988
38.5 Intel® Trace Hub ..... 989
38.6 On-Board Debug Port Connector ..... 989

**Volume 3: Electrical, Mechanical, and Thermal ..... 990**

**39 Signal Pin Names and Signal MUXing ..... 991**
39.1 Directory of Signal Names and Pin Names ..... 993
39.1.1 Reserved (RSVD_xxx) Pins ..... 1026
39.2 Directory of Signals that Share the HSIO Pins ..... 1028
39.3 Directory of Voltage Supply and Sense Pin Names ..... 1029
39.4 Directory of Rotation-Detect Pins ..... 1032
39.5 Directory of All Signal and Voltage Pins - Sorted by Pin Number ..... 1033

**40 Signal Electrical and Timing Characteristics ..... 1044**
40.1 DDR Memory Interface ..... 1044
40.1.1 DDR4 ..... 1044
40.1.1.1 Deviations From The DDR4 Standard ..... 1044
40.1.1.2 Supplementary DDR4 Signals Not Covered By The DDR4 Standard ..... 1044
40.2 PCI Express Root Port Interface ..... 1045
40.2.1 Deviations From The PCI Express 3.0 Standard ..... 1045
40.2.2 Supplementary PCI Express Signals Not Covered By The PCI Express 3.0 Standard ..... 1045
40.3 SATA Interface ..... 1046
40.3.1 Deviations From The SATA 3.1 Standard ..... 1046
40.3.2 Supplementary SATA Signals Not Covered By The SATA 3.1 Standard ..... 1046
40.4 USB Interface ..... 1047
40.4.1 Deviations From The USB 3.0 Standard ..... 1047
40.4.2 Supplementary USB Signals Not Covered By The USB 3.0 Standard ..... 1047
40.4.3 USB 2.0 Standard ..... 1047
40.5 SMBus Interface ..... 1048
40.5.1 Deviations From The SMBus 2.0 Standard ..... 1048
40.5.2 Supplementary SMBus Signals Not Covered By The SMBus 2.0 Standard ..... 1048
40.6 LPC Interface ..... 1049
40.6.1 Deviations From The LPC 1.1 Standard ..... 1049
40.6.2 Supplementary LPC Signals Not Covered By The LPC 1.1 Standard ..... 1049
40.7 eMMC Interface ..... 1050
40.7.1 Deviations From The eMMC 5.0 Standard ..... 1050
40.7.2 Supplementary eMMC Signals Not Covered By The eMMC 5.0 Standard ..... 1050
40.8 HSUART Interface ..... 1051
40.8.1 Deviations From Texas Instruments PC16550D Standard ..... 1051
40.8.2 Supplementary HSUART Signals Not Covered By The Texas Instruments PC16550D Standard ..... 1051
40.9 Serial Peripheral Interface (SPI) ..... 1052
40.9.1 Interface Timing Parameters and Waveforms ..... 1052
40.9.2 SPI Electrical Characteristics ..... 1057
40.10 LAN Interface ..... 1058
40.10.1 10GBASE-KR Interface ..... 1058
40.10.1.1 Deviations From The IEEE802.3 10GBASE-KR Standard ..... 1058

40.10.1.2 Supplementary 10GBASE-KR Signals Not Covered By The IEEE802.3 10GBASE-KR Standard ..... 1058
40.10.2 SFI Interface ..... 1058
40.10.2.1 Deviations From The SFF-8431 SFI Standard ..... 1058
40.10.2.2 Supplementary SFI Signals Not Covered By The SFF-8431 SFI Standard ..... 1058
40.10.3 2500BASE-X Interface ..... 1058
40.10.3.1 Deviations From The 2500BASE-X Industry Convention ..... 1058
40.10.3.2 Supplementary 2500BASE-X Signals Not Covered By The 2500BASE-X Industry Convention ..... 1058
40.10.4 1000BASE-KX Interface ..... 1059
40.10.4.1 Deviations From The IEEE802.3 1000BASE-KX Standard ..... 1059
40.10.4.2 Supplementary 1000BASE-KX Signals Not Covered By The IEEE802.3 1000BASE-KX Standard ..... 1059
40.10.5 SGMII Interface ..... 1059
40.10.5.1 Deviations From The Cisco Systems SGMII Industry Standard ..... 1059
40.10.5.2 Supplementary SGMII Signals Not Covered By The Cisco Systems SGMII Industry Standard ..... 1059
40.10.6 Management Data Input/Output (MDIO) ..... 1060
40.10.6.1 Deviations From IEEE 802.3 ..... 1060
40.10.6.2 Supplementary MDIO Signals Not Covered By Clause 22 and Clause 45 of IEEE 802.3 ..... 1060
40.10.7 Slave ("Target") Network Controller Sideband Interface (NC-SI) ..... 1060
40.10.7.1 Deviations From The NC-SI Standard ..... 1060
40.10.7.2 Supplementary NC-SI Signals Not Covered By The NC-SI Standard ..... 1060
40.11 SoC Serial Voltage Identification (SVID) Interface ..... 1061
40.11.1 SoC SVID Timing Deviates from the VR13.0 Specification ..... 1061
40.11.2 Supplementary Signals Not Covered By The SVID Standard ..... 1061
40.12 SoC JTAG and Debug Interfaces ..... 1061
40.12.1 Deviations From The JTAG Standard ..... 1061
40.12.2 Supplementary Signals Not Covered By The JTAG Standard ..... 1061
40.13 Customer General Purpose I/O (GPIO) Interface ..... 1062
40.13.1 1.05V GPIO (CMOS-LV Buffer Type) Electrical Characteristics ..... 1062
40.13.1.1 Receiver Electrical Characteristics ..... 1062
40.13.1.2 Driver Electrical Characteristics ..... 1063
40.13.2 1.8V GPIO (CMOS-MV Buffer Type) Electrical Characteristics ..... 1064
40.13.2.1 Receiver Electrical Characteristics ..... 1064
40.13.2.2 Driver Electrical Characteristics ..... 1065
40.13.3 3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics ..... 1066
40.13.3.1 Receiver Electrical Characteristics ..... 1066
40.13.3.2 Driver Electrical Characteristics ..... 1067
40.14 Miscellaneous Interfaces ..... 1068
40.14.1 Signals of "V Group V1P05" (CMOS-LV Buffer Type) ..... 1068
40.14.2 Signals of "V Group V1P80" (CMOS-MV Buffer Type) ..... 1068
40.14.3 Signals of "V Group V3P30" (CMOS-HV Buffer Type) ..... 1068

**41 Operating Conditions and Power Requirements ..... 1069**
41.1 Absolute Maximum and Minimum Ratings ..... 1069
41.1.1 Component Storage Conditions Specification ..... 1069
41.1.1.1 Prior to Board-Attach ..... 1069
41.1.1.2 Post Board-Attach ..... 1070
41.2 Normal Operating Conditions ..... 1070
41.2.1 Component Temperature ..... 1070
41.2.2 Supply Voltage and Current Requirements ..... 1071

**42 Package Mechanical Overview ..... 1076**

**Volume 4: Registers ..... 1077**

**43 Introduction ..... 1078**
43.1 Access to Registers in Host Root Space ..... 1078
43.1.1 PCI Configuration Space ..... 1078
43.1.1.1 Unimplemented Devices/Functions and Registers ..... 1078
43.1.2 Memory Space—Relocatable Address ..... 1079
43.1.3 Memory Space—Fixed Address ..... 1079
43.1.4 I/O Space—Relocatable Address ..... 1079
43.1.5 I/O Space—Fixed Address ..... 1079
43.1.6 Sideband Registers ..... 1080
43.1.7 Model-Specific Registers (MSR) ..... 1080
43.2 Register Description Terminology ..... 1080
43.2.1 Default Values ..... 1080
43.2.2 Software Access Attributes ..... 1081
43.2.3 Hexadecimal, Binary and Decimal Numbers ..... 1082

**44 System Agent - B0, D0, F0 ..... 1083**
44.1 Introduction and Index ..... 1083
44.1.1 Host Configuration Space ..... 1084
44.1.2 Host Memory Space—MCHBAR ..... 1085
44.1.3 Host Memory Space—DEFVTDBAR ..... 1096
44.1.4 Host Memory Space—DEFVTDBAR ..... 1097
44.1.5 Sideband Registers ..... 1099
44.2 Registers in Configuration Space ..... 1103
44.2.1 Device ID and Vendor ID Register (DEVICE_ID_VENDOR_ID_0_0_0_PCI)—Offset 0h ..... 1103
44.2.2 PCI Status and PCI Command Register (PCI_STATUS_COMMAND_0_0_0_PCI)—Offset 4h ..... 1103
44.2.3 PCI Revision ID and PCI Class Code Register (REVISION_ID_CLASS_CODE_0_0_0_PCI)—Offset 8h ..... 1104
44.2.4 Master Latency Timer and Header Type Register (MASTER_LATENCY_TIME_0_0_0_PCI)—Offset Ch ..... 1104
44.2.5 PCI Subsystem Vendor ID and PCI Subsystem ID (SVID_SID_0_0_0_PCI)—Offset 2Ch ..... 1105
44.2.6 Capability Register Pointer (CAPPTR_0_0_0_PCI)—Offset 34h ..... 1105
44.2.7 Memory Controller Hub Base Address Register (MCHBAR_LO_0_0_0_PCI)—Offset 48h ..... 1106
44.2.8 Memory Controller Hub Base Address Register (MCHBAR_HI_0_0_0_PCI)—Offset 4Ch ..... 1106
44.2.9 Device Enable Register (DEVEN_0_0_0_PCI)—Offset 54h ..... 1107
44.2.10 PCI Express Enhanced Configuration Range Base Address Low (PCIEXBAR_LO_0_0_0_PCI)—Offset 60h ..... 1108
44.2.11 PCI Express Enhanced Configuration Range Base Address High (PCIEXBAR_HI_0_0_0_PCI)—Offset 64h ..... 1109
44.2.12 Top of Upper Usable DRAM Low (TOUUD_LO_0_0_0_PCI)—Offset A8h ..... 1110
44.2.13 Top of Upper Usable DRAM High (TOUUD_HI_0_0_0_PCI)—Offset ACh ..... 1111
44.2.14 Base of Data Stolen Memory (BDSM_0_0_0_PCI)—Offset B0h ..... 1112
44.2.15 Base of Graphics Stolen Memory (BGSM_0_0_0_PCI)—Offset B4h ..... 1113
44.2.16 Top Segment Memory Base (TSEGMB_0_0_0_PCI)—Offset B8h ..... 1114
44.2.17 Top of Lower Usable DRAM (TOLUD_0_0_0_PCI)—Offset BCh ..... 1115
44.2.18 Scratchpad (SKPD_0_0_0_PCI)—Offset DCh ..... 1115
44.2.19 Capability ID0 Capability Control (CAPID0_CAPCTRL0_0_0_0_PCI)—Offset E0h ..... 1116
44.2.20 Capability ID0 A (CAPID0_A_0_0_0_PCI)—Offset E4h ..... 1117

44.2.21 Capability ID0 B (CAPID0_B_0_0_0_PCI)—Offset E8h ..... 1118
44.2.22 Manufacturer ID Register (MANUFACTURER_ID_0_0_0_PCI)—Offset F4h . 1119
44.3 Registers in Memory Space—MCHBAR ..... 1120
44.3.1 Default VTd Base Address Register (DEFVTDBAR_0_0_0_MCHBAR_C)—Offset 6C80h ..... 1120
44.3.2 Spare BIOS (A_CR_SPARE_BIOS_MCHBAR)—Offset 647Ch ..... 1121
44.3.3 Uncorrectable Error Status Register (A_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6588h..... 1122
44.3.4 Uncorrectable Error Mask Register (A_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 658Ch ..... 1125
44.3.5 Uncorrectable Error Severity Register (A_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6590h..... 1127
44.3.6 First Uncorrectable Error Register (A_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6594h ..... 1129
44.3.7 Next Uncorrectable Error Register (A_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6598h..... 1132
44.3.8 Uncorrectable Error Select Register (A_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 659Ch ..... 1135
44.3.9 Uncorrectable Error Count (A_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 65A0h ..... 1137
44.3.10 Correctable Error Status Register (A_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 65A4h..... 1138
44.3.11 Correctable Error Mask Register (A_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 65A8h ..... 1141
44.3.12 First Correctable Error Register (A_CR_CORFERRSTS_0_0_0_MCHBAR)—Offset 65ACh ..... 1143
44.3.13 Next Correctable Error Register (A_CR_CORNERRSTS_0_0_0_MCHBAR)—Offset 65B0h..... 1146
44.3.14 Correctable Error Count Select (A_CR_CORERRCNTSEL_0_0_0_MCHBAR)—Offset 65B4h ..... 1149
44.3.15 Correctable Error Count (A_CR_CORERRCNT_0_0_0_MCHBAR)—Offset 65B8h ..... 1151
44.3.16 Slice and Channel Hash (A_CR_SLICE_CHANNEL_HASH_0_0_0_MCHBAR)—Offset 65C0h..... 1152
44.3.17 A_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR—Offset 65D0h ..... 1154
44.3.18 A_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR—Offset 65D4h ..... 1154
44.3.19 B-Unit Miscellaneous Configuration (B_CR_BMISC_0_0_0_MCHBAR)—Offset 6800h..... 1155
44.3.20 Slice 0 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE0)—Offset 6868h ..... 1155
44.3.21 Slice 1 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE1)—Offset 686Ch ..... 1155
44.3.22 IMR0 Base (B_CR_BIMR0BASE_0_0_0_MCHBAR)—Offset 6870h..... 1156
44.3.23 IMR0 Mask (B_CR_BIMR0MASK_0_0_0_MCHBAR)—Offset 6874h..... 1157
44.3.24 IMR0 Control Policy (B_CR_BIMR0CP_0_0_0_MCHBAR)—Offset 6878h .... 1157
44.3.25 IMR0 Read Access Policy (B_CR_BIMR0RAC_0_0_0_MCHBAR)—Offset 6880h..... 1158
44.3.26 IMR0 Write Access Policy (B_CR_BIMR0WAC_0_0_0_MCHBAR)—Offset 6888h..... 1161
44.3.27 B_CR_BIMR1BASE_0_0_0_MCHBAR—Offset 6890h..... 1164
44.3.28 IMR1 Mask (B_CR_BIMR1MASK_0_0_0_MCHBAR)—Offset 6894h..... 1165
44.3.29 IMR1 Control Policy (B_CR_BIMR1CP_0_0_0_MCHBAR)—Offset 6898h .... 1165
44.3.30 IMR1 Read Access Policy (B_CR_BIMR1RAC_0_0_0_MCHBAR)—Offset 68A0h ..... 1166
44.3.31 IMR1 Write Access Policy (B_CR_BIMR1WAC_0_0_0_MCHBAR)—Offset 68A8h ..... 1169
44.3.32 Base 0 IMR2 Base (B_CR_BIMR2BASE_0_0_0_MCHBAR)—Offset 68B0h .. 1172

44.3.33 IMR2 Mask (B_CR_BIMR2MASK_0_0_0_MCHBAR)—Offset 68B4h .....1173
44.3.34 IMR2 Control Policy (B_CR_BIMR2CP_0_0_0_MCHBAR)—Offset 68B8h.....1173
44.3.35 IMR2 Read Access Policy (B_CR_BIMR2RAC_0_0_0_MCHBAR)—Offset 68C0h .....1174
44.3.36 IMR2 Write Access Policy (B_CR_BIMR2WAC_0_0_0_MCHBAR)—Offset 68C8h .....1177
44.3.37 IMR3 Base (B_CR_BIMR3BASE_0_0_0_MCHBAR)—Offset 68D0h.....1180
44.3.38 IMR3 Mask (B_CR_BIMR3MASK_0_0_0_MCHBAR)—Offset 68D4h.....1181
44.3.39 IMR3 Control Policy (B_CR_BIMR3CP_0_0_0_MCHBAR)—Offset 68D8h ....1181
44.3.40 IMR3 Read Access Policy (B_CR_BIMR3RAC_0_0_0_MCHBAR)—Offset 68E0h .....1182
44.3.41 IMR3 Write Access Policy (B_CR_BIMR3WAC_0_0_0_MCHBAR)—Offset 68E8h .....1185
44.3.42 IMR4 Base (B_CR_BIMR4BASE_0_0_0_MCHBAR)—Offset 68F0h .....1188
44.3.43 IMR4 Mask (B_CR_BIMR4MASK_0_0_0_MCHBAR)—Offset 68F4h .....1189
44.3.44 B-Unit IMR4 Control Policy (B_CR_BIMR4CP_0_0_0_MCHBAR)—Offset 68F8h .....1189
44.3.45 IMR4 Read Access Policy (B_CR_BIMR4RAC_0_0_0_MCHBAR)—Offset 6900h .....1190
44.3.46 IMR4 Write Access Policy (B_CR_BIMR4WAC_0_0_0_MCHBAR)—Offset 6908h .....1193
44.3.47 IMR5 Base (B_CR_BIMR5BASE_0_0_0_MCHBAR)—Offset 6910h .....1196
44.3.48 IMR5 Mask (B_CR_BIMR5MASK_0_0_0_MCHBAR)—Offset 6914h .....1197
44.3.49 IMR5 Control Policy (B_CR_BIMR5CP_0_0_0_MCHBAR)—Offset 6918h.....1197
44.3.50 IMR5 Read Access Policy (B_CR_BIMR5RAC_0_0_0_MCHBAR)—Offset 6920h .....1198
44.3.51 IMR5 Write Access Policy (B_CR_BIMR5WAC_0_0_0_MCHBAR)—Offset 6928h .....1201
44.3.52 IMR6 Base (B_CR_BIMR6BASE_0_0_0_MCHBAR)—Offset 6930h .....1204
44.3.53 IMR6 Mask (B_CR_BIMR6MASK_0_0_0_MCHBAR)—Offset 6934h .....1205
44.3.54 IMR6 Control Policy (B_CR_BIMR6CP_0_0_0_MCHBAR)—Offset 6938h.....1205
44.3.55 IMR6 Read Access Policy (B_CR_BIMR6RAC_0_0_0_MCHBAR)—Offset 6940h .....1206
44.3.56 IMR6 Write Access Policy (B_CR_BIMR6WAC_0_0_0_MCHBAR)—Offset 6948h .....1209
44.3.57 IMR7 Base (B_CR_BIMR7BASE_0_0_0_MCHBAR)—Offset 6950h .....1212
44.3.58 IMR7 Mask (B_CR_BIMR7MASK_0_0_0_MCHBAR)—Offset 6954h .....1213
44.3.59 IMR7 Control Policy (B_CR_BIMR7CP_0_0_0_MCHBAR)—Offset 6958h.....1213
44.3.60 IMR7 Read Access Policy (B_CR_BIMR7RAC_0_0_0_MCHBAR)—Offset 6960h .....1214
44.3.61 IMR7 Write Access Policy (B_CR_BIMR7WAC_0_0_0_MCHBAR)—Offset 6968h .....1217
44.3.62 IMR8 Base (B_CR_BIMR8BASE_0_0_0_MCHBAR)—Offset 6970h .....1220
44.3.63 IMR8 Mask (B_CR_BIMR8MASK_0_0_0_MCHBAR)—Offset 6974h .....1221
44.3.64 IMR8 Control Policy (B_CR_BIMR8CP_0_0_0_MCHBAR)—Offset 6978h.....1221
44.3.65 IMR8 Read Access Policy (B_CR_BIMR8RAC_0_0_0_MCHBAR)—Offset 6980h .....1222
44.3.66 IMR8 Write Access Policy (B_CR_BIMR8WAC_0_0_0_MCHBAR)—Offset 6988h .....1225
44.3.67 IMR9 Base (B_CR_BIMR9BASE_0_0_0_MCHBAR)—Offset 6990h .....1228
44.3.68 IMR9 Mask (B_CR_BIMR9MASK_0_0_0_MCHBAR)—Offset 6994h .....1229
44.3.69 IMR9 Control Policy (B_CR_BIMR9CP_0_0_0_MCHBAR)—Offset 6998h.....1229
44.3.70 IMR9 Read Access Policy (B_CR_BIMR9RAC_0_0_0_MCHBAR)—Offset 69A0h .....1230
44.3.71 IMR9 Write Access Policy (B_CR_BIMR9WAC_0_0_0_MCHBAR)—Offset 69A8h .....1233

44.3.72 IMR10 Base (B_CR_BIMR10BASE_0_0_0_MCHBAR)—Offset 69B0h ..... 1236
44.3.73 B_CR_BIMR10MASK_0_0_0_MCHBAR—Offset 69B4h ..... 1237
44.3.74 IMR10 Control Policy (B_CR_BIMR10CP_0_0_0_MCHBAR)—Offset 69B8h. 1237
44.3.75 IMR10 Read Access Policy (B_CR_BIMR10RAC_0_0_0_MCHBAR)—Offset 69C0h ..... 1238
44.3.76 IMR10 Write Access Policy (B_CR_BIMR10WAC_0_0_0_MCHBAR)—Offset 69C8h ..... 1241
44.3.77 IMR11 Base (B_CR_BIMR11BASE_0_0_0_MCHBAR)—Offset 69D0h ..... 1244
44.3.78 IMR11 Mask (B_CR_BIMR11MASK_0_0_0_MCHBAR)—Offset 69D4h ..... 1245
44.3.79 IMR11 Control Policy (B_CR_BIMR11CP_0_0_0_MCHBAR)—Offset 69D8h 1245
44.3.80 IMR11 Read Access Policy (B_CR_BIMR11RAC_0_0_0_MCHBAR)—Offset 69E0h..... 1246
44.3.81 IMR11 Write Access Policy (B_CR_BIMR11WAC_0_0_0_MCHBAR)—Offset 69E8h..... 1249
44.3.82 IMR12 Base (B_CR_BIMR12BASE_0_0_0_MCHBAR)—Offset 69F0h ..... 1252
44.3.83 IMR12 Mask (B_CR_BIMR12MASK_0_0_0_MCHBAR)—Offset 69F4h ..... 1253
44.3.84 IMR12 Control Policy (B_CR_BIMR12CP_0_0_0_MCHBAR)—Offset 69F8h. 1253
44.3.85 IMR12 Read Access Policy (B_CR_BIMR12RAC_0_0_0_MCHBAR)—Offset 6A00h ..... 1254
44.3.86 IMR12 Write Access Policy (B_CR_BIMR12WAC_0_0_0_MCHBAR)—Offset 6A08h ..... 1257
44.3.87 IMR13 Base (B_CR_BIMR13BASE_0_0_0_MCHBAR)—Offset 6A10h ..... 1260
44.3.88 IMR13 Mask (B_CR_BIMR13MASK_0_0_0_MCHBAR)—Offset 6A14h ..... 1261
44.3.89 IMR13 Control Policy (B_CR_BIMR13CP_0_0_0_MCHBAR)—Offset 6A18h. 1261
44.3.90 IMR13 Read Access Policy (B_CR_BIMR13RAC_0_0_0_MCHBAR)—Offset 6A20h ..... 1262
44.3.91 IMR13 Write Access Policy (B_CR_BIMR13WAC_0_0_0_MCHBAR)—Offset 6A28h ..... 1265
44.3.92 IMR14 Base (B_CR_BIMR14BASE_0_0_0_MCHBAR)—Offset 6A30h ..... 1268
44.3.93 IMR14 Mask (B_CR_BIMR14MASK_0_0_0_MCHBAR)—Offset 6A34h ..... 1269
44.3.94 IMR14 Control Policy (B_CR_BIMR14CP_0_0_0_MCHBAR)—Offset 6A38h. 1269
44.3.95 IMR14 Read Access Policy (B_CR_BIMR14RAC_0_0_0_MCHBAR)—Offset 6A40h ..... 1270
44.3.96 IMR14 Write Access Policy (B_CR_BIMR14WAC_0_0_0_MCHBAR)—Offset 6A48h ..... 1273
44.3.97 IMR15 Base (B_CR_BIMR15BASE_0_0_0_MCHBAR)—Offset 6A50h ..... 1276
44.3.98 IMR15 Mask (B_CR_BIMR15MASK_0_0_0_MCHBAR)—Offset 6A54h ..... 1277
44.3.99 IMR15 Control Policy (B_CR_BIMR15CP_0_0_0_MCHBAR)—Offset 6A58h. 1277
44.3.100 IMR15 Read Access Policy (B_CR_BIMR15RAC_0_0_0_MCHBAR)—Offset 6A60h ..... 1278
44.3.101 IMR15 Write Access Policy (B_CR_BIMR15WAC_0_0_0_MCHBAR)—Offset 6A68h ..... 1281
44.3.102 IMR16 Base (B_CR_BIMR16BASE_0_0_0_MCHBAR)—Offset 6A70h ..... 1284
44.3.103 IMR16 Mask (B_CR_BIMR16MASK_0_0_0_MCHBAR)—Offset 6A74h ..... 1285
44.3.104 IMR16 Control Policy (B_CR_BIMR16CP_0_0_0_MCHBAR)—Offset 6A78h. 1285
44.3.105 IMR16 Read Access Policy (B_CR_BIMR16RAC_0_0_0_MCHBAR)—Offset 6A80h ..... 1286
44.3.106 IMR16 Write Access Policy (B_CR_BIMR16WAC_0_0_0_MCHBAR)—Offset 6A88h ..... 1289
44.3.107 IMR17 Base (B_CR_BIMR17BASE_0_0_0_MCHBAR)—Offset 6A90h ..... 1292
44.3.108 IMR17 Mask (B_CR_BIMR17MASK_0_0_0_MCHBAR)—Offset 6A94h ..... 1293
44.3.109 IMR17 Control Policy (B_CR_BIMR17CP_0_0_0_MCHBAR)—Offset 6A98h. 1293
44.3.110 IMR17 Read Access Policy (B_CR_BIMR17RAC_0_0_0_MCHBAR)—Offset 6AA0h ..... 1294
44.3.111 IMR17 Write Access Policy (B_CR_BIMR17WAC_0_0_0_MCHBAR)—Offset 6AA8h ..... 1297

44.3.112 IMR18 Base (B_CR_BIMR18BASE_0_0_0_MCHBAR)—Offset 6AB0h .....1300
44.3.113 IMR18 Mask (B_CR_BIMR18MASK_0_0_0_MCHBAR)—Offset 6AB4h .....1301
44.3.114 IMR18 Control Policy (B_CR_BIMR18CP_0_0_0_MCHBAR)—Offset 6AB8h .1301
44.3.115 IMR18 Read Access Policy (B_CR_BIMR18RAC_0_0_0_MCHBAR)—Offset 6AC0h .....1302
44.3.116 IMR18 Write Access Policy (B_CR_BIMR18WAC_0_0_0_MCHBAR)—Offset 6AC8h .....1305
44.3.117 IMR19 Base (B_CR_BIMR19BASE_0_0_0_MCHBAR)—Offset 6AD0h .....1308
44.3.118 IMR19 Mask (B_CR_BIMR19MASK_0_0_0_MCHBAR)—Offset 6AD4h .....1309
44.3.119 IMR19 Control Policy (B_CR_BIMR19CP_0_0_0_MCHBAR)—Offset 6AD8h.1309
44.3.120 IMR19 Read Access Policy (B_CR_BIMR19RAC_0_0_0_MCHBAR)—Offset 6AE0h .....1310
44.3.121 IMR19 Write Access Policy (B_CR_BIMR19WAC_0_0_0_MCHBAR)—Offset 6AE8h .....1313
44.3.122 TPM Selector (B_CR_TPM_SELECTOR_0_0_0_MCHBAR)—Offset 6C24h ....1316
44.3.123 Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR)—Offset 6C80h.....1316
44.3.124 B-Unit Arbiter Control BARBCTRL0 (B_CR_BARBCTRL0)—Offset 6D4Ch ....1317
44.3.125 B-Unit Arbiter Control BARBCTRL1 (B_CR_BARBCTRL1)—Offset 6D50h ....1318
44.3.126 B-Unit Scheduler Control (B_CR_BSCHWT0)—Offset 6D54h .....1319
44.3.127 B-Unit Scheduler Control (B_CR_BSCHWT1)—Offset 6D58h .....1320
44.3.128 B-Unit Scheduler Control (B_CR_BSCHWT2)—Offset 6D5Ch .....1321
44.3.129 B-Unit Scheduler Control (B_CR_BSCHWT3)—Offset 6D60h .....1322
44.3.130 B-Unit Flush Control (B_CR_BWFLUSH)—Offset 6D64h.....1323
44.3.131 B-Unit Flush Weights (B_CR_BFLWT)—Offset 6D68h .....1323
44.3.132 Weighted Scheduling Control of High Priority ISOC and Other Requests (B_CR_BISOCWT)—Offset 6D6Ch .....1324
44.3.133 B-Unit Control (B_CR_BCTRL2)—Offset 6D70h.....1325
44.3.134 Uncorrectable Error Status Register (B_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6E04h .....1326
44.3.135 Uncorrectable Error Mask Register (B_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 6E08h.....1329
44.3.136 Uncorrectable Error Severity Register (B_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6E0Ch .....1331
44.3.137 First Uncorrectable Error Register (B_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6E10h.....1333
44.3.138 Next Uncorrectable Error Register (B_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6E14h .....1336
44.3.139 Uncorrectable Error Select Register (B_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 6E18h .....1339
44.3.140 Uncorrectable Error Count (B_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 6E1Ch .....1341
44.3.141 Correctable Error Status Register (B_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 6E20h .....1342
44.3.142 Correctable Error Mask Register (B_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 6E24h.....1345
44.3.143 Data Parity Header Log Low (B_CR_DPHDRLOG_LO_0_0_0_MCHBAR)—Offset 6E28h .....1347
44.3.144 Data Parity Header Log High (B_CR_DPHDRLOG_HI_0_0_0_MCHBAR)—Offset 6E2Ch .....1347
44.3.145 Data Parity Address Log Low (B_CR_DPADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E30h .....1347
44.3.146 Data Parity Address Log High (B_CR_DPADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E34h.....1348
44.3.147 D-Unit Error Address Log (B_CR_DERRADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E38h .....1348

44.3.148 D-Unit Error Address Log (B_CR_DERRADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E3Ch ..... 1348
44.3.149 Asymmetric Memory Region 0 (B_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR)—Offset 6E40h ..... 1349
44.3.150 Asymmetric Memory Region 1 (B_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR)—Offset 6E44h ..... 1349
44.3.151 B-Unit Machine Check Mode Low (B_CR_BMCMODE_LOW)—Offset 6E48h..... 1350
44.3.152 B-Unit Machine Check Mode High (B_CR_BMCMODE_HIGH)—Offset 6E4Ch ..... 1350
44.3.153 PWR_LIMIT_MISC_0_0_0_MCHBAR (P_CR_PWR_LIMIT_MISC_0_0_0_MCHBAR)—Offset 7014h ..... 1351
44.3.154 CAPABILITY_ID_0_0_0_MCHBAR (P_CR_CAPABILITY_ID_0_0_0_MCHBAR)—Offset 7018h ..... 1351
44.3.155 MC_BIOS_REQ_0_0_0_MCH (P_CR_MC_BIOS_REQ_0_0_0_MCHBAR)—Offset 7020h ..... 1352
44.3.156 MCA_ERROR_SRC_0_0_0_MCHBAR (P_CR_MCA_ERROR_SRC_0_0_0_MCHBAR)—Offset 702Ch ..... 1353
44.3.157 DDR_THERM_THRT_CTRL_0_0_0_MCHBAR (P_CR_DDR_THERM_THRT_CTRL_0_0_0_MCHBAR)—Offset 7030h ..... 1354
44.3.158 PACKAGE_THERM_MARGIN (P_CR_PACKAGE_THERM_MARGIN_0_0_0_MCHBAR)—Offset 703Ch ..... 1355
44.3.159 DDR_RAPL_LIMIT (P_CR_DDR_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 7040h..... 1356
44.3.160 DDR_ENERGY_STATUS (P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7048h ..... 1357
44.3.161 DDR_RAPL_PERF_STATUS (P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 704Ch..... 1357
44.3.162 PACKAGE_RAPL_PERF_STATUS (P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 7050h.. 1358
44.3.163 PRIMARY_PLANE_TURBO_POWER_POLICY (P_CR_PRIMARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7054h.. 1358
44.3.164 SECONDARY_PLANE_TURBO_POWER_POLICY (P_CR_SECONDARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7058h..... 1359
44.3.165 PRIMARY_PLANE_ENERGY_STATUS (P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 705Ch ..... 1359
44.3.166 SECONDARY_PLANE_ENERGY_STATUS (P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7060h..... 1360
44.3.167 PACKAGE_POWER_SKU_UNIT (P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR)—Offset 7068h.... 1361
44.3.168 PACKAGE_ENERGY_STATUS (P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 706Ch ..... 1362
44.3.169 GT_PERF_STATUS (P_CR_GT_PERF_STATUS_0_0_0_MCHBAR)—Offset 7070h..... 1362
44.3.170 TEMPERATURE_TARGET (P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR)—Offset 7074h ..... 1363
44.3.171 BIOS_RESET_CPL (P_CR_BIOS_RESET_CPL_0_0_0_MCHBAR)—Offset 7078h..... 1364
44.3.172 BIOS_MAILBOX_DATA (P_CR_BIOS_MAILBOX_DATA_0_0_0_MCHBAR)—Offset 7080h ..... 1366
44.3.173 BIOS_MAILBOX_INTERFACE (P_CR_BIOS_MAILBOX_INTERFACE_0_0_0_MCHBAR)—Offset 7084h ..... 1366

44.3.174 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7088h .....1367
44.3.175 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 708Ch .....1368
44.3.176 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7090h .....1369
44.3.177 FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7094h .....1369
44.3.178 NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7098h .....1370
44.3.179 PACKAGE_POWER_SKU (P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR)—Offset 70A0h.....1371
44.3.180 PACKAGE_RAPL_LIMIT (P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 70A8h .....1372
44.3.181 IA_PERF_LIMIT_REASONS (P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR)—Offset 70B0h .....1374
44.3.182 MEMHOT_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_STATUS_0_0_0_MCHBAR)—Offset 70D8h.....1376
44.3.183 MEMHOT_THERM_CONFIG_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_CONFIG_0_0_0_MCHBAR)—Offset 70DCh .....1376
44.3.184 MEM_THERM_CTRL_0_0_0_MCHBAR (P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR)—Offset 7200h .....1377
44.3.185 MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR)—Offset 7204h.....1378
44.3.186 MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR)—Offset 7208h....1378
44.3.187 MEM_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR)—Offset 720Ch.....1379
44.3.188 MEM_THERM_INTERRUPT_0_0_0_MCHBAR (P_CR_MEM_THERM_INTERRUPT_0_0_0_MCHBAR)—Offset 7210h .....1380
44.3.189 MEM_THERM_INT_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_INT_STATUS_0_0_0_MCHBAR)—Offset 7214h .....1381
44.3.190 MEM_THERM_TEMP_OFST_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_OFST_0_0_0_MCHBAR)—Offset 7218h .....1383
44.3.191 TELEM_NEAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_NEAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E0h .1384
44.3.192 TELEM_FAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_FAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E8h....1384
44.3.193 TURBO_ACTIVATION_RATIO (P_CR_TURBO_ACTIVATION_RATIO_0_0_0_MCHBAR)—Offset 70F0h .....1385
44.3.194 C2C3TT_CFG (P_CR_C2C3TT_CFG_0_0_0_MCHBAR)—Offset 7034h .....1385
44.3.195 DTS_CONFIG1_CFG (P_CR_DTS_CONFIG1_CFG_0_0_0_MCHBAR)—Offset 7110h .....1386
44.3.196 DTS_CONFIG2_CFG (P_CR_DTS_CONFIG2_CFG_0_0_0_MCHBAR)—Offset 7114h .....1386
44.3.197 DTS_CONFIG3_CFG (P_CR_DTS_CONFIG3_CFG_0_0_0_MCHBAR)—Offset 7118h .....1387
44.3.198 SOUTH_IO_PORT_STATUS3 (P_CR_SOUTH_IO_PORT_STATUS3_0_0_0_MCHBAR)—Offset 711Ch.....1387
44.3.199 DRAM_BIOS_INFO (P_CR_DRAM_BIOS_INFO_0_0_0_MCHBAR)—Offset 7128h .....1388
44.3.200 ADR_COMMAND (P_CR_ADR_COMMAND_0_0_0_MCHBAR)—Offset 712Ch1389

44.3.201 DRAM_POWER_INFO (P_CR_DRAM_POWER_INFO_0_0_0_MCHBAR)—Offset 7130h ..... 1390
44.3.202 PCODE_WRITE_SPARE (P_CR_PCODE_WRITE_SPARE_0_0_0_MCHBAR)—Offset 7138h..... 1390
44.3.203 PACKAGE_TEMPERATURES_0_0_0_MCHBAR (P_CR_PACKAGE_TEMPERATURES_0_0_0_MCHBAR)—Offset 70F4h ..... 1391
44.3.204 MEMSS_FREQUENCY_CAPABILITIES (P_CR_MEMSS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7108h..... 1392
44.3.205 PP1_C0_CORE_CLOCK_0_0_0_MCHBAR (P_CR_PP1_C0_CORE_CLOCK_0_0_0_MCHBAR)—Offset 7160h..... 1392
44.3.206 Core Exists Vector (P_CR_CORE_EXISTS_VECTOR_0_0_0_MCHBAR)—Offset 7164h ..... 1393
44.3.207 Software Core Disable Mask (P_CR_CORE_DISABLE_MASK_0_0_0_MCHBAR)—Offset 7168h..... 1394
44.3.208 PL3_CONTROL_MCHBAR (P_CR_PL3_CONTROL_0_0_0_MCHBAR)—Offset 71F0h..... 1396
44.3.209 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL1_0_0_0_MCHBAR)—Offset 7220h..... 1398
44.3.210 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL2_0_0_0_MCHBAR)—Offset 7228h..... 1399
44.3.211 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL3_0_0_0_MCHBAR)—Offset 7230h..... 1400
44.3.212 CONFIG_TDP_CONTROL (P_CR_CONFIG_TDP_CONTROL_0_0_0_MCHBAR)—Offset 7238h..... 1400
44.3.213 CONFIG_TDP_NOMINAL (P_CR_CONFIG_TDP_NOMINAL_0_0_0_MCHBAR)—Offset 723Ch..... 1401
44.3.214 Graphics Superqueue Active Clocks (P_CR_PP1_ANY_THREAD_ACTIVITY_0_0_0_MCHBAR)—Offset 7244h..... 1402
44.3.215 B-Unit Copy of Default VTd BAR PMEN (B_CR_PMEN_REG_0_0_0_DEFVTDBAR)—Offset 64h ..... 1402
44.3.216 B-Unit Copy of Default VTd BAR PLM Base Register (B_CR_PLMBASE_REG_0_0_0_DEFVTDBAR)—Offset 68h ..... 1403
44.3.217 B-Unit Copy of Default VTd BAR PLM Limit Register (B_CR_PLMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 6Ch ..... 1404
44.3.218 B-Unit Copy of Default VTd BAR PHM Base Register (B_CR_PHMBASE_REG_0_0_0_DEFVTDBAR)—Offset 70h ..... 1405
44.3.219 B-Unit Copy of Default VTd BAR PHM Limit Register (B_CR_PHMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 78h ..... 1406
44.4 Registers in Memory Space—DEFVTDBAR..... 1407
44.4.1 Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h ..... 1407
44.4.2 Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h ..... 1407
44.4.3 Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h... 1409
44.4.4 Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h ..... 1412
44.4.5 Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch ..... 1413
44.4.6 Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h ..... 1414
44.4.7 Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h ... 1415
44.4.8 Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h..... 1417
44.4.9 Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h .. 1419
44.4.10 Fault Event Data Register (FEDATA_REG_0_0_0_VTDBAR)—Offset 3Ch ... 1420
44.4.11 Fault Event Address Register (FEADDR_REG_0_0_0_VTDBAR)—Offset 40h ..... 1420
44.4.12 Fault Event Upper Address Register (FEUADDR_REG_0_0_0_VTDBAR)—Offset 44h ..... 1420
44.4.13 Advanced Fault Log Register (AFLOG_REG_0_0_0_VTDBAR)—Offset 58h . 1421

44.4.14 Protected Memory Enable Register (PMEN_REG_0_0_0_VTDBAR)—Offset 64h....1422
44.4.15 Protected Low Memory Base Register (PLMBASE_REG_0_0_0_VTDBAR)—Offset 68h....1423
44.4.16 Protected Low-Memory Limit Register (PLMLIMIT_REG_0_0_0_VTDBAR)—Offset 6Ch....1424
44.4.17 Protected High-Memory Base Register (PHMBASE_REG_0_0_0_VTDBAR)—Offset 70h....1425
44.4.18 Protected High-Memory Limit Register (PHMLIMIT_REG_0_0_0_VTDBAR)—Offset 78h....1426
44.4.19 Invalidation Queue Head Register (IQH_REG_0_0_0_VTDBAR)—Offset 80h....1427
44.4.20 Invalidation Queue Tail Register (IQT_REG_0_0_0_VTDBAR)—Offset 88h....1427
44.4.21 Invalidation Queue Address Register (IQA_REG_0_0_0_VTDBAR)—Offset 90h....1428
44.4.22 Invalidation Completion Status Register (ICS_REG_0_0_0_VTDBAR)—Offset 9Ch....1428
44.4.23 Invalidation Event Control Register (IECTL_REG_0_0_0_VTDBAR)—Offset A0h....1429
44.4.24 Invalidation Event Data Register (IEDATA_REG_0_0_0_VTDBAR)—Offset A4h....1430
44.4.25 Invalidation Event Address Register (IEADDR_REG_0_0_0_VTDBAR)—Offset A8h....1430
44.4.26 Invalidation Event Upper Address Register (IEUADDR_REG_0_0_0_VTDBAR)—Offset ACh....1431
44.4.27 Interrupt Remapping Table Address Register (IRTA_REG_0_0_0_VTDBAR)—Offset B8h....1431
44.4.28 Page Request Status Register (PRESTS_REG_0_0_0_VTDBAR)—Offset DCh....1432
44.4.29 Page Request Event Control Register (PRECTL_REG_0_0_0_VTDBAR)—Offset E0h....1433
44.4.30 Page Request Event Data Register (PREDATA_REG_0_0_0_VTDBAR)—Offset E4h....1434
44.4.31 Page Request Event Address Register (PREADDR_REG_0_0_0_VTDBAR)—Offset E8h....1434
44.4.32 Page Request Event Upper Address Register (PREUADDR_REG_0_0_0_VTDBAR)—Offset ECh....1434
44.4.33 Fault Recording Register Low [0] (FRCDL_REG_0_0_0_VTDBAR)—Offset 400h....1435
44.4.34 Fault Recording Register High [0] (FRCDH_REG_0_0_0_VTDBAR)—Offset 408h....1436
44.4.35 Invalidate Address Register (IVA_REG_0_0_0_VTDBAR)—Offset 500h....1438
44.4.36 IOTLB Invalidate Register (IOTLB_REG_0_0_0_VTDBAR)—Offset 508h....1439
44.5 Sideband Registers....1441
44.5.1 UCELOG—Offset 64h....1441
44.5.2 UCEADDR—Offset 68h....1442
44.5.3 SBELOG—Offset 6Ch....1442
44.5.4 SBEADDR—Offset 70h....1443
44.5.5 SBECNT0—Offset 74h....1444
44.5.6 SBECNT1—Offset 78h....1444
44.5.7 SBECNT2—Offset 7Ch....1445
44.5.8 SBECNT3—Offset 80h....1445
44.5.9 SBEACC0—Offset 84h....1445
44.5.10 SBEACC1—Offset 88h....1446
44.5.11 SBEACC2—Offset 8Ch....1446
44.5.12 SBEACC3—Offset 90h....1446

44.5.13 DPATROL_SCRUB_CFG—Offset B0h ..... 1447
44.5.14 UCELOG1—Offset CCh ..... 1447
44.5.15 SBELOG1—Offset D0h ..... 1448
44.5.16 CORRERRTHRSHLD0—Offset 100h ..... 1448
44.5.17 CORRERRTHRSHLD1—Offset 104h ..... 1448
44.5.18 CORRERRTHRSHLD2—Offset 108h ..... 1449
44.5.19 CORRERRTHRSHLD3—Offset 10Ch ..... 1449
44.5.20 LEAKY_BUCKET_CFG0—Offset 110h ..... 1449
44.5.21 LEAKY_BUCKET_CFG1—Offset 114h ..... 1450
44.5.22 LEAKY_BUCKET_CFG2—Offset 118h ..... 1450
44.5.23 LEAKY_BUCKET_CFG3—Offset 11Ch ..... 1450
44.5.24 FERRNERR—Offset 120h ..... 1451
44.5.25 DERRSTS—Offset 124h ..... 1452
44.5.26 DERRMSKSEV—Offset 128h ..... 1453
44.5.27 DERRCNTSEL—Offset 12Ch ..... 1454
44.5.28 DERRCNT—Offset 130h ..... 1454
44.5.29 ERRINJCTL—Offset 134h ..... 1455
44.5.30 ERRINJADDR—Offset 138h ..... 1456
44.5.31 ERRINJAMSK—Offset 13Ch ..... 1456
44.5.32 ERRINJDATA0—Offset 140h ..... 1456
44.5.33 ERRINJDATA1—Offset 144h ..... 1456
44.5.34 DCS—Offset 170h ..... 1457
44.5.35 DECCCTRL—Offset 180h ..... 1458
44.5.36 LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch ..... 1459
44.5.37 DPATROL_SCRUB_TMR—Offset 194h ..... 1459
44.5.38 UCELOG—Offset 64h ..... 1460
44.5.39 UCEADDR—Offset 68h ..... 1460
44.5.40 SBELOG—Offset 6Ch ..... 1461
44.5.41 SBEADDR—Offset 70h ..... 1461
44.5.42 SBECNT0—Offset 74h ..... 1462
44.5.43 SBECNT1—Offset 78h ..... 1462
44.5.44 SBECNT2—Offset 7Ch ..... 1463
44.5.45 SBECNT3—Offset 80h ..... 1463
44.5.46 SBEACC0—Offset 84h ..... 1464
44.5.47 SBEACC1—Offset 88h ..... 1464
44.5.48 SBEACC2—Offset 8Ch ..... 1465
44.5.49 SBEACC3—Offset 90h ..... 1465
44.5.50 DPATROL_SCRUB_CFG—Offset B0h ..... 1466
44.5.51 UCELOG1—Offset CCh ..... 1466
44.5.52 SBELOG1—Offset D0h ..... 1467
44.5.53 CORRERRTHRSHLD0—Offset 100h ..... 1467
44.5.54 CORRERRTHRSHLD1—Offset 104h ..... 1467
44.5.55 CORRERRTHRSHLD2—Offset 108h ..... 1468
44.5.56 CORRERRTHRSHLD3—Offset 10Ch ..... 1468
44.5.57 LEAKY_BUCKET_CFG0—Offset 110h ..... 1468
44.5.58 LEAKY_BUCKET_CFG1—Offset 114h ..... 1469
44.5.59 LEAKY_BUCKET_CFG2—Offset 118h ..... 1469
44.5.60 LEAKY_BUCKET_CFG3—Offset 11Ch ..... 1469
44.5.61 FERRNERR—Offset 120h ..... 1470
44.5.62 DERRSTS—Offset 124h ..... 1471
44.5.63 DERRMSKSEV—Offset 128h ..... 1472
44.5.64 DERRCNTSEL—Offset 12Ch ..... 1473
44.5.65 DERRCNT—Offset 130h ..... 1473
44.5.66 ERRINJCTL—Offset 134h ..... 1474
44.5.67 ERRINJADDR—Offset 138h ..... 1475

44.5.68 ERRINJAMSK—Offset 13Ch ..... 1475
44.5.69 ERRINJDATA0—Offset 140h ..... 1475
44.5.70 ERRINJDATA1—Offset 144h ..... 1475
44.5.71 DCS—Offset 170h ..... 1476
44.5.72 DECCCTRL—Offset 180h ..... 1478
44.5.73 LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch ..... 1479
44.5.74 DPATROL_SCRUB_TMR—Offset 194h ..... 1479
44.5.75 ICLK_DWORD8—Port 99h, Offset 220h ..... 1480

**45 Global and Local Error Registers (GLMREG) - B0, D4, F0 ..... 1481**
45.1 Introduction and Index ..... 1481
45.1.1 Host Configuration Space ..... 1482
45.2 Registers in Configuration Space ..... 1483
45.2.1 GLREG Vendor ID (GLREG_VID)—Offset 0h ..... 1483
45.2.2 GLREG PCI Command (GLREG_PCICMD)—Offset 4h ..... 1484
45.2.3 GLREG PCI Status (GLREG_PCISTS)—Offset 6h ..... 1485
45.2.4 GLREG Class Code (GLREG_CC)—Offset 9h ..... 1486
45.2.5 GLREG Cacheline Size (GLREG_CLS)—Offset Ch ..... 1486
45.2.6 GLREG Subsystem Vendor ID (GLREG_SVID)—Offset 2Ch ..... 1487
45.2.7 GLREG Capabilities Pointer (GLREG_CAPPTR)—Offset 34h ..... 1487
45.2.8 GLREG Interrupt Pin (GLREG_INTP)—Offset 3Dh ..... 1488
45.2.9 GLREG PCI Express Capability List (GLREG_EXPCAPLST)—Offset 40h ..... 1488
45.2.10 GLREG PCI Express Capabilities (GLREG_EXPCAP)—Offset 42h ..... 1489
45.2.11 GLREG Device Control (GLREG_DEVCTL)—Offset 48h ..... 1490
45.2.12 GLREG Device Status (GLREG_DEVSTS)—Offset 4Ah ..... 1491
45.2.13 GLREG Global Correctable Error Status (GLREG_GCOERRSTS)—Offset 200h ..... 1492
45.2.14 GLREG Global Non-Fatal Error Status (GLREG_GNFERRSTS)—Offset 204h ..... 1494
45.2.15 GLREG Global Fatal Error Status (GLREG_GFAERRSTS)—Offset 208h ..... 1496
45.2.16 GLREG Global Error Mask (GLREG_GERRMSK)—Offset 20Ch ..... 1498
45.2.17 GLREG Global Error Timer (GLREG_GTIME)—Offset 228h ..... 1500
45.2.18 GLREG Global Correctable FERR Error Time Stamp (GLREG_GCOFERRTIME)—Offset 230h ..... 1500
45.2.19 GLREG Global Nonfatal FERR Error Time Stamp (GLREG_GNFFERRTIME)—Offset 238h ..... 1501
45.2.20 GLREG Global Fatal FERR Error Time Stamp (GLREG_GFAFERRTIME)—Offset 240h ..... 1501
45.2.21 GLREG Global System Event Status (GLREG_GSYSEVTSTS)—Offset 248h ..... 1502
45.2.22 GLREG Global System Event Mask (GLREG_GSYSEVTMSK)—Offset 24Ch ..... 1502
45.2.23 GLREG Global System Event Map (GLREG_GSYSEVTMAP)—Offset 250h ..... 1503
45.2.24 GLREG Error Pin Control (GLREG_ERRPINCTRL)—Offset 254h ..... 1504
45.2.25 GLREG Error Pin Status (GLREG_ERRPINSTS)—Offset 258h ..... 1504
45.2.26 GLREG Error Pin Data (GLREG_ERRPINDATA)—Offset 25Ch ..... 1505
45.2.27 GLREG Uncorrectable Error Counter (GLREG_LUNCERRCNT)—Offset 294h ..... 1506

**46 Root Complex Event Collector - B0, D5, F0 ..... 1507**
46.1 Introduction and Index ..... 1507
46.1.1 Host Configuration Space ..... 1508
46.2 Registers in Configuration Space ..... 1509
46.2.1 RCEC Vendor ID (RCEC_VID)—Offset 0h ..... 1509
46.2.2 RCEC PCI Command (RCEC_PCICMD)—Offset 4h ..... 1510
46.2.3 RCEC PCI Status (RCEC_PCISTS)—Offset 6h ..... 1511

46.2.4 RCEC Class Code (RCEC_CC)—Offset 9h ..... 1512
46.2.5 RCEC Cacheline Size (RCEC_CLS)—Offset Ch ..... 1512
46.2.6 RCEC Interrupt Pin (RCEC_INTP)—Offset 3Dh ..... 1513
46.2.7 RCEC PCI Express Capability List (RCEC_EXPCAPLST)—Offset 40h ..... 1513
46.2.8 RCEC PCI Express Capabilities (RCEC_EXPCAP)—Offset 42h ..... 1514
46.2.9 RCEC Device Control (RCEC_DEVCTL)—Offset 48h ..... 1515
46.2.10 RCEC Device Status (RCEC_DEVSTS)—Offset 4Ah ..... 1516
46.2.11 RCEC Root Control (RCEC_ROOTCTL)—Offset 5Ch ..... 1517
46.2.12 RCEC Power Management Capabilities (RCEC_PMCAP)—Offset 82h ..... 1518
46.2.13 RCEC Power Management Control / Status (RCEC_PMCSR)—Offset 84h ..... 1518
46.2.14 RCEC MSI Capability List (RCEC_MSICAPLST)—Offset 90h ..... 1519
46.2.15 RCEC MSI Message Control (RCEC_MSICTL)—Offset 92h ..... 1519
46.2.16 RCEC MSI Message Address (RCEC_MSIADDR)—Offset 94h ..... 1520
46.2.17 RCEC MSI Message Data (RCEC_MSIDATA)—Offset 98h ..... 1520
46.2.18 RCEC MSI Mask Bit (RCEC_MSIMSK)—Offset 9Ch ..... 1521
46.2.19 RCEC MSI Pending Bit (RCEC_MSIPENDING)—Offset A0h ..... 1521
46.2.20 RCEC Advanced Error Reporting Extended Capability Header (RCEC_AERCAPHDR)—Offset 100h ..... 1521
46.2.21 RCEC Uncorrectable Error Status (RCEC_ERRUNCSTS)—Offset 104h ..... 1522
46.2.22 RCEC Correctable Error Status (RCEC_ERRCORSTS)—Offset 110h ..... 1523
46.2.23 RCEC Correctable Error Mask (RCEC_ERRCORMSK)—Offset 114h ..... 1524
46.2.24 RCEC Advanced Error Capabilities and Control (RCEC_AERCAPCTL)—Offset 118h ..... 1525
46.2.25 RCEC Header Log (RCEC_AERHDRLOG1)—Offset 11Ch ..... 1525
46.2.26 RCEC Header Log (RCEC_AERHDRLOG2)—Offset 120h ..... 1526
46.2.27 RCEC Header Log (RCEC_AERHDRLOG3)—Offset 124h ..... 1526
46.2.28 RCEC Header Log (RCEC_AERHDRLOG4)—Offset 128h ..... 1526
46.2.29 RCEC Root Error Command (RCEC_ROOTERRCMD)—Offset 12Ch ..... 1527
46.2.30 RCEC Root Error Status (RCEC_ROOTERRSTS)—Offset 130h ..... 1528
46.2.31 RCEC Error Source Identification (RCEC_ERRSRCID)—Offset 134h ..... 1529

**47 RP - Intel® QuickAssist Technology - B0, D6/D22/D23, F0 ..... 1530**
47.1 Introduction and Index ..... 1530
47.1.1 Host Configuration Space ..... 1531
47.2 Registers in Configuration Space ..... 1534
47.2.1 VID—Offset 0h ..... 1534
47.2.2 DID—Offset 2h ..... 1534
47.2.3 PCICMD—Offset 4h ..... 1535
47.2.4 PCISTS—Offset 6h ..... 1536
47.2.5 CCR—Offset 9h ..... 1537
47.2.6 CLS—Offset Ch ..... 1537
47.2.7 IOBASE—Offset 1Ch ..... 1538
47.2.8 IOLIMIT—Offset 1Dh ..... 1538
47.2.9 MEMBASE—Offset 20h ..... 1539
47.2.10 MEMLIMIT—Offset 22h ..... 1540
47.2.11 PFBASE—Offset 24h ..... 1541
47.2.12 PFLIMIT—Offset 26h ..... 1542
47.2.13 PFBASEU—Offset 28h ..... 1542
47.2.14 PFLIMITU—Offset 2Ch ..... 1543
47.2.15 CAPPTR—Offset 34h ..... 1543
47.2.16 INTL—Offset 3Ch ..... 1544
47.2.17 INTP—Offset 3Dh ..... 1544
47.2.18 BCTL—Offset 3Eh ..... 1544
47.2.19 EXPCAPLST—Offset 40h ..... 1546

47.2.20 EXPCAP—Offset 42h ..... 1547
47.2.21 DEVCAP—Offset 44h ..... 1548
47.2.22 DEVCTL—Offset 48h ..... 1549
47.2.23 DEVSTS—Offset 4Ah ..... 1550
47.2.24 LINKCAP—Offset 4Ch ..... 1551
47.2.25 LINKCTL—Offset 50h ..... 1553
47.2.26 LINKSTS—Offset 52h ..... 1554
47.2.27 SLOTSTS—Offset 5Ah ..... 1555
47.2.28 ROOTCTL—Offset 5Ch ..... 1556
47.2.29 ROOTCAP—Offset 5Eh ..... 1557
47.2.30 ROOTSTS—Offset 60h ..... 1557
47.2.31 DEVCAP2—Offset 64h ..... 1558
47.2.32 DEVCTL2—Offset 68h ..... 1560
47.2.33 LINKCAP2—Offset 6Ch ..... 1561
47.2.34 LINKCTL2—Offset 70h ..... 1562
47.2.35 LINKSTS2—Offset 72h ..... 1564
47.2.36 PMCAPLST—Offset 80h ..... 1564
47.2.37 PMCAP—Offset 82h ..... 1565
47.2.38 PMCSR—Offset 84h ..... 1566
47.2.39 PMBSE—Offset 86h ..... 1566
47.2.40 SSCAPLST—Offset 88h ..... 1567
47.2.41 SSVID—Offset 8Ch ..... 1568
47.2.42 SSID—Offset 8Eh ..... 1568
47.2.43 MSICAPLST—Offset 90h ..... 1569
47.2.44 MSICTL—Offset 92h ..... 1569
47.2.45 MSIADDR—Offset 94h ..... 1570
47.2.46 MSIDATA—Offset 98h ..... 1570
47.2.47 MSIMSK—Offset 9Ch ..... 1571
47.2.48 MSIPENDING—Offset A0h ..... 1571
47.2.49 RPPMCSR—Offset A8h ..... 1572
47.2.50 PMCSRS—Offset B0h ..... 1572
47.2.51 SMICSR—Offset C8h ..... 1573
47.2.52 SCICSR—Offset CCh ..... 1573
47.2.53 PBTXNCTL—Offset D8h ..... 1574
47.2.54 PLKCTL—Offset EAh ..... 1574
47.2.55 LTRCSR—Offset ECh ..... 1575
47.2.56 LTRL—Offset F0h ..... 1576
47.2.57 LTROVR—Offset F4h ..... 1577
47.2.58 INTXSWZCTL—Offset F8h ..... 1578
47.2.59 AERCAPCTL—Offset 118h ..... 1579
47.2.60 AERHDRLOG1—Offset 11Ch ..... 1579
47.2.61 AERHDRLOG2—Offset 120h ..... 1580
47.2.62 AERHDRLOG3—Offset 124h ..... 1580
47.2.63 AERHDRLOG4—Offset 128h ..... 1580
47.2.64 ROOTERRCMD—Offset 12Ch ..... 1581
47.2.65 ROOTERRSTS—Offset 130h ..... 1581
47.2.66 ERRSRCID—Offset 134h ..... 1582
47.2.67 ACSCAPHDR—Offset 138h ..... 1582
47.2.68 ACSCAP—Offset 13Ch ..... 1583
47.2.69 ACSCTL—Offset 13Eh ..... 1584
47.2.70 XPTDEF—Offset 2DCh ..... 1585
47.2.71 XPTDEF2—Offset 2E0h ..... 1587
47.2.72 IOSFDEVCLKGCTL—Offset 2E4h ..... 1589
47.2.73 SBDEVCLKGCTL—Offset 2E6h ..... 1589
47.2.74 IDFDEVCLKGCTL—Offset 2E8h ..... 1590

**48 PCI Express Root Ports - B0, D(9-12, 14-17), F0 ..... 1591**
48.1 Introduction and Index ..... 1591
48.1.1 Host Configuration Space ..... 1592
48.1.2 Sideband Registers ..... 1599
48.2 Registers in Configuration Space ..... 1600
48.2.1 VID—Offset 0h ..... 1600
48.2.2 DID—Offset 2h ..... 1600
48.2.3 PCICMD—Offset 4h ..... 1601
48.2.4 PCISTS—Offset 6h ..... 1602
48.2.5 RID—Offset 8h ..... 1603
48.2.6 CCR—Offset 9h ..... 1603
48.2.7 CLS—Offset Ch ..... 1604
48.2.8 PLAT—Offset Dh ..... 1604
48.2.9 HDR—Offset Eh ..... 1604
48.2.10 BIST—Offset Fh ..... 1605
48.2.11 PRIBUS—Offset 18h ..... 1605
48.2.12 SECBUS—Offset 19h ..... 1606
48.2.13 SUBBUS—Offset 1Ah ..... 1606
48.2.14 IOBASE—Offset 1Ch ..... 1607
48.2.15 IOLIMIT—Offset 1Dh ..... 1607
48.2.16 SECSTS—Offset 1Eh ..... 1608
48.2.17 MEMBASE—Offset 20h ..... 1609
48.2.18 MEMLIMIT—Offset 22h ..... 1610
48.2.19 PFBASE—Offset 24h ..... 1611
48.2.20 PFLIMIT—Offset 26h ..... 1612
48.2.21 PFBASEU—Offset 28h ..... 1612
48.2.22 PFLIMITU—Offset 2Ch ..... 1613
48.2.23 CAPPTR—Offset 34h ..... 1613
48.2.24 INTL—Offset 3Ch ..... 1614
48.2.25 INTP—Offset 3Dh ..... 1614
48.2.26 BCTL—Offset 3Eh ..... 1615
48.2.27 EXPCAPLST—Offset 40h ..... 1617
48.2.28 EXPCAP—Offset 42h ..... 1617
48.2.29 DEVCAP—Offset 44h ..... 1618
48.2.30 DEVCTL—Offset 48h ..... 1619
48.2.31 DEVSTS—Offset 4Ah ..... 1620
48.2.32 LINKCAP—Offset 4Ch ..... 1621
48.2.33 LINKCTL—Offset 50h ..... 1623
48.2.34 LINKSTS—Offset 52h ..... 1625
48.2.35 SLOTCAP—Offset 54h ..... 1626
48.2.36 SLOTCTL—Offset 58h ..... 1628
48.2.37 SLOTSTS—Offset 5Ah ..... 1630
48.2.38 ROOTCTL—Offset 5Ch ..... 1632
48.2.39 ROOTCAP—Offset 5Eh ..... 1633
48.2.40 ROOTSTS—Offset 60h ..... 1633
48.2.41 DEVCAP2—Offset 64h ..... 1634
48.2.42 DEVCTL2—Offset 68h ..... 1637
48.2.43 LINKCAP2—Offset 6Ch ..... 1639
48.2.44 LINKCTL2—Offset 70h ..... 1640
48.2.45 LINKSTS2—Offset 72h ..... 1642
48.2.46 PMCAPLST—Offset 80h ..... 1642
48.2.47 PMCAP—Offset 82h ..... 1643
48.2.48 PMCSR—Offset 84h ..... 1644
48.2.49 PMBSE—Offset 86h ..... 1644
48.2.50 SSCAPLST—Offset 88h ..... 1645

48.2.51 SSVID—Offset 8Ch....1646
48.2.52 SSID—Offset 8Eh....1646
48.2.53 MSICAPLST—Offset 90h....1646
48.2.54 MSICTL—Offset 92h....1647
48.2.55 MSIADDR—Offset 94h....1647
48.2.56 MSIDATA—Offset 98h....1648
48.2.57 MSIMSK—Offset 9Ch....1649
48.2.58 MSIPENDING—Offset A0h....1649
48.2.59 RPPMCSR—Offset A8h....1650
48.2.60 PMCSRS—Offset B0h....1652
48.2.61 PPBIFCTL—Offset C0h....1652
48.2.62 SMICSR—Offset C8h....1653
48.2.63 SCICSR—Offset CCh....1654
48.2.64 SSCTL—Offset D0h....1655
48.2.65 EXPPTMBARCTL—Offset D2h....1656
48.2.66 PPD0—Offset D4h....1657
48.2.67 PBTXNCTL—Offset D8h....1659
48.2.68 PBPHYCTL—Offset E0h....1660
48.2.69 PLKCTL—Offset EAh....1660
48.2.70 LTRCSR—Offset ECh....1661
48.2.71 LTRL—Offset F0h....1662
48.2.72 LTROVR—Offset F4h....1663
48.2.73 INTXSWZCTL—Offset F8h....1664
48.2.74 CFGAGTERR—Offset FCh....1665
48.2.75 AERCAPHDR—Offset 100h....1666
48.2.76 ERRUNCSTS—Offset 104h....1667
48.2.77 ERRUNCMSK—Offset 108h....1669
48.2.78 ERRUNCSEV—Offset 10Ch....1671
48.2.79 ERRCORSTS—Offset 110h....1673
48.2.80 ERRCORMSK—Offset 114h....1674
48.2.81 AERCAPCTL—Offset 118h....1675
48.2.82 AERHDRLOG1—Offset 11Ch....1676
48.2.83 AERHDRLOG2—Offset 120h....1676
48.2.84 AERHDRLOG3—Offset 124h....1676
48.2.85 AERHDRLOG4—Offset 128h....1677
48.2.86 ROOTERRCMD—Offset 12Ch....1677
48.2.87 ROOTERRSTS—Offset 130h....1678
48.2.88 ERRSRCID—Offset 134h....1679
48.2.89 ACSCAPHDR—Offset 138h....1679
48.2.90 ACSCAP—Offset 13Ch....1680
48.2.91 ACSCTL—Offset 13Eh....1681
48.2.92 ERRUNCDETMSK—Offset 140h....1682
48.2.93 ERRCORDETMSK—Offset 144h....1683
48.2.94 ROOTERRDETMSK—Offset 148h....1683
48.2.95 MCSTCAPHDR—Offset 150h....1684
48.2.96 MCSTCAP—Offset 154h....1684
48.2.97 MCSTCTL—Offset 156h....1684
48.2.98 MCSTBAR—Offset 158h....1685
48.2.99 MCSTUBAR—Offset 15Ch....1685
48.2.100 MCSTRCV—Offset 160h....1685
48.2.101 MCSTBLKALL—Offset 168h....1686
48.2.102 MCSTOLBAR—Offset 178h....1686
48.2.103 MCSTUOLBAR—Offset 17Ch....1686
48.2.104 EINJCAPHDR—Offset 180h....1687
48.2.105 EINJHDR—Offset 184h....1687

48.2.106 EINJCTL—Offset 188h ..... 1688
48.2.107 SECEXPCAP—Offset 200h ..... 1689
48.2.108 LINKCTL3—Offset 204h ..... 1690
48.2.109 LANEERRSTS—Offset 208h ..... 1691
48.2.110 LANEEQCTL0—Offset 20Ch ..... 1692
48.2.111 LANEEQCTL1—Offset 20Eh ..... 1694
48.2.112 LANEEQCTL2—Offset 210h ..... 1696
48.2.113 LANEEQCTL3—Offset 212h ..... 1698
48.2.114 LANEEQCTL4—Offset 214h ..... 1700
48.2.115 LANEEQCTL5—Offset 216h ..... 1702
48.2.116 LANEEQCTL6—Offset 218h ..... 1704
48.2.117 LANEEQCTL7—Offset 21Ah ..... 1706
48.2.118 XPPMDL0—Offset 280h ..... 1708
48.2.119 XPPMDL1—Offset 284h ..... 1708
48.2.120 XPPMCL0—Offset 288h ..... 1708
48.2.121 XPPMCL1—Offset 28Ch ..... 1708
48.2.122 XPPMDH—Offset 290h ..... 1709
48.2.123 XPPMCH—Offset 292h ..... 1709
48.2.124 XPPMR0—Offset 294h ..... 1710
48.2.125 XPPMR1—Offset 298h ..... 1713
48.2.126 XPPMEVL0—Offset 29Ch ..... 1716
48.2.127 XPPMEVL1—Offset 2A0h ..... 1718
48.2.128 XPPMEVH0—Offset 2A4h ..... 1720
48.2.129 XPPMEVH1—Offset 2A8h ..... 1721
48.2.130 XPPMER0—Offset 2ACh ..... 1722
48.2.131 XPPMER1—Offset 2B0h ..... 1724
48.2.132 XPPMDFXMAT0—Offset 2B4h ..... 1726
48.2.133 XPPMDFXMAT1—Offset 2B8h ..... 1727
48.2.134 XPPMDFXMSK0—Offset 2BCh ..... 1728
48.2.135 XPPMDFXMSK1—Offset 2C0h ..... 1728
48.2.136 XPPERFCON—Offset 2C4h ..... 1729
48.2.137 XPPERFSTAT—Offset 2C8h ..... 1730
48.2.138 XPCDTHROTTLEP—Offset 2D0h ..... 1730
48.2.139 XPCDTHROTTLEN—Offset 2D4h ..... 1731
48.2.140 XPCDTHROTTLEC—Offset 2D8h ..... 1731
48.2.141 IOSFDEVCLKGCTL—Offset 2E4h ..... 1732
48.2.142 SBDEVCLKGCTL—Offset 2E6h ..... 1732
48.2.143 VDHDRCAP—Offset 300h ..... 1733
48.2.144 IBSTHDRCAP—Offset 304h ..... 1733
48.2.145 IBSTENGLKSPDCAP—Offset 310h ..... 1734
48.2.146 IBSTENGLKSPDCTL—Offset 314h ..... 1735
48.2.147 IBSTENGLKTXRXCFG—Offset 31Ch ..... 1736
48.2.148 IBSTERRCRCSTS0—Offset 320h ..... 1736
48.2.149 IBSTERRRCRVSTS0—Offset 324h ..... 1737
48.2.150 IBSTERRCRCSTS1—Offset 328h ..... 1737
48.2.151 IBSTERRRCRVSTS1—Offset 32Ch ..... 1738
48.2.152 IBSTERRCRCSTS2—Offset 330h ..... 1738
48.2.153 IBSTERRRCRVSTS2—Offset 334h ..... 1739
48.2.154 IBSTERRCRCSTS3—Offset 338h ..... 1739
48.2.155 IBSTERRRCRVSTS3—Offset 33Ch ..... 1740
48.2.156 IBSTPHYLCTL0—Offset 344h ..... 1741
48.2.157 IBSTPHYLCTL1—Offset 348h ..... 1742
48.2.158 IBSTPHYLCTL2—Offset 34Ch ..... 1743
48.2.159 IBSTPHYLCTL3—Offset 350h ..... 1744
48.2.160 IBSTTXLCTL—Offset 354h ..... 1745

48.2.161 IBSTTDSTS—Offset 358h ....1745
48.2.162 IBSTPLLNSTS0—Offset 36Ch ....1746
48.2.163 IBSTPRECFG0—Offset 370h....1747
48.2.164 IBSTPOSTCFG0—Offset 374h....1748
48.2.165 IBSTPLLNSTS1—Offset 378h ....1749
48.2.166 IBSTPRECFG1—Offset 37Ch ....1750
48.2.167 IBSTPOSTCFG1—Offset 380h....1751
48.2.168 TLPCIECLKGCTL—Offset 8E0h....1751
48.2.169 TLPCIECLKGCCTL—Offset 8E2h....1752
48.2.170 PT0TXNCLKGCTL—Offset 8E4h....1752
48.2.171 PT1TXNCLKGCTL—Offset 8E6h....1753
48.2.172 PT2TXNCLKGCTL—Offset 8E8h....1753
48.2.173 PT3TXNCLKGCTL—Offset 8EAh ....1754
48.2.174 XPTDEF2—Offset 8F8h....1755
48.2.175 PMIDLTMR—Offset 948h ....1757
48.2.176 LLRBSIZE—Offset 98Ch ....1758
48.2.177 FRMERRMSK—Offset 99Ch ....1759
48.2.178 FRMERRSTS—Offset 99Eh ....1760
48.2.179 LLPCIECLKGCTL—Offset 9E0h....1760
48.2.180 LLPCIECLKGCCTL—Offset 9E2h....1761
48.2.181 PT0LNKCLKGCTL—Offset 9E4h....1761
48.2.182 PT1LNKCLKGCTL—Offset 9E6h....1762
48.2.183 PT2LNKCLKGCTL—Offset 9E8h....1762
48.2.184 PT3LNKCLKGCTL—Offset 9EAh....1763
48.2.185 CFGPCIECLKGCTL—Offset 9ECh ....1763
48.2.186 CFGPCIECLKGCCTL—Offset 9EEh ....1763
48.2.187 LTSSMSMSTS—Offset A68h....1764
48.2.188 LTSSMLNSTS0—Offset A70h....1765
48.2.189 LTSSMSTATELOG0—Offset A80h ....1765
48.2.190 LTSSMSTATELOG1—Offset A84h ....1765
48.2.191 LTSSMSTATELOG2—Offset A88h ....1766
48.2.192 LTSSMSTATELOGCTL—Offset A8Ch ....1766
48.2.193 LTLNERRCTRL—Offset A94h ....1766
48.2.194 LTLNERRSTS—Offset A96h ....1767
48.2.195 LTLNERRLOG—Offset A98h....1768
48.2.196 UPCFGCSR—Offset AB8h....1769
48.2.197 SOSCTL—Offset AC8h....1770
48.2.198 G3SOSERRSTS—Offset ACCh....1770
48.2.199 PLPCIECLKGCTL—Offset AE0h....1770
48.2.200 PLPCIECLKGCCTL—Offset AE2h....1771
48.2.201 PT0PHYCLKGCTL—Offset AE4h....1771
48.2.202 PT1PHYCLKGCTL—Offset AE6h....1772
48.2.203 PT2PHYCLKGCTL—Offset AE8h....1772
48.2.204 PT3PHYCLKGCTL—Offset AEAh....1773
48.2.205 LTLGCAP—Offset B00h ....1773
48.2.206 LTLGCTRL—Offset B04h....1774
48.2.207 LTLGSTART0—Offset B08h ....1775
48.2.208 LTLGSTART1—Offset B0Ch ....1776
48.2.209 LTLGSTART2—Offset B10h ....1777
48.2.210 LTLGSTART3—Offset B14h ....1778
48.2.211 LTLGSTP0—Offset B18h ....1779
48.2.212 LTLGSTP1—Offset B1Ch....1779
48.2.213 LTLGSTP2—Offset B20h ....1780
48.2.214 LTLGSTP3—Offset B24h ....1780
48.2.215 LTLGTMEV0—Offset B28h....1781

48.2.216 LTLGTMEV1—Offset B2Ch ..... 1782
48.2.217 LTLGTMEV2—Offset B30h ..... 1783
48.2.218 LTLGTMEV3—Offset B34h ..... 1784
48.2.219 LTLGRDCT—Offset B40h ..... 1784
48.2.220 LTLGRES—Offset B44h ..... 1785
48.2.221 LTLGSMEV0—Offset B48h ..... 1785
48.2.222 LTLGSMEV1—Offset B4Ch ..... 1786
48.2.223 NFTSCOMCLK—Offset B50h ..... 1786
48.2.224 NFTSDSCCLK—Offset B54h ..... 1787
48.2.225 LEKBERR—Offset B58h ..... 1787
48.2.226 EXPBERR0—Offset B5Ch ..... 1787
48.2.227 EXPBERR1—Offset B60h ..... 1788
48.2.228 LEKBPROERR—Offset B64h ..... 1788
48.2.229 SPEEDCTL—Offset B68h ..... 1789
48.2.230 RCVRDETSTS—Offset B6Ch ..... 1791
48.2.231 RCVRERR—Offset B70h ..... 1791
48.2.232 EBERR—Offset B74h ..... 1792
48.2.233 EBERRMSK—Offset B78h ..... 1792
48.2.234 OBEINJCTL—Offset B7Ch ..... 1793
48.2.235 LTSSMERRSTS0—Offset B80h ..... 1795
48.2.236 LTSSMERRSTS1—Offset B84h ..... 1797
48.2.237 EQEVALCTL6—Offset B8Ch ..... 1798
48.2.238 EQEVALCTL7—Offset B90h ..... 1799
48.2.239 EQFARFSLF—Offset B9Ch ..... 1800
48.2.240 FOMINCCTL—Offset BA4h ..... 1801
48.2.241 PHYRECAL—Offset BA8h ..... 1801
48.2.242 PLLSTBYCTL—Offset BACh ..... 1802
48.2.243 PLUTXEQCTL—Offset BB0h ..... 1802
48.2.244 PHYCGIDLEMSK—Offset BB4h ..... 1803
48.2.245 EQEVALCTL4—Offset BB8h ..... 1804
48.2.246 EQEVALCTL5—Offset BBCh ..... 1805
48.2.247 LEKBLNERRCNT0—Offset BC0h ..... 1805
48.2.248 LEKBLNERRCNT1—Offset BC4h ..... 1806
48.2.249 EQEVALCTL—Offset BD0h ..... 1806
48.2.250 EQEVALCTL2—Offset BD4h ..... 1807
48.2.251 EQEVALCTL3—Offset BD8h ..... 1808
48.2.252 EQEVALSTS—Offset BDCh ..... 1809
48.2.253 EQEVALSTS2—Offset BE0h ..... 1811
48.2.254 EQEVALSTS3—Offset BE4h ..... 1812
48.2.255 G3FRAMEERR—Offset BECh ..... 1814
48.3 Sideband Registers - Devices 9-12 ..... 1815
48.3.1 PPBIFCTL_PRIV—Offset 10h ..... 1815
48.4 Sideband Registers - Devices 14-17 ..... 1816
48.4.1 PPBIFCTL_PRIV—Offset 10h ..... 1816

**49 SMBus - Host - B0, D18, F0 ..... 1817**
49.1 Introduction and Index ..... 1817
49.1.1 Host Configuration Space ..... 1818
49.1.2 Host Memory Space—SMTBAR ..... 1819
49.2 Registers in Configuration Space ..... 1820
49.2.1 VID—Offset 0h ..... 1820
49.2.2 DID—Offset 2h ..... 1820
49.2.3 PCICMD—Offset 4h ..... 1821
49.2.4 PCISTS—Offset 6h ..... 1822
49.2.5 RID—Offset 8h ..... 1823

49.2.6 CCR—Offset 9h ..... 1823
49.2.7 SMTBAR—Offset 10h ..... 1824
49.2.8 SVID—Offset 2Ch ..... 1824
49.2.9 SID—Offset 2Eh ..... 1825
49.2.10 INTL—Offset 3Ch ..... 1825
49.2.11 INTP—Offset 3Dh ..... 1826
49.2.12 EXPCAPLST—Offset 40h ..... 1826
49.2.13 DEVCAP—Offset 44h ..... 1827
49.2.14 DEVCTL—Offset 48h ..... 1829
49.2.15 DEVSTS—Offset 4Ah ..... 1830
49.2.16 PMCAPLST—Offset 80h ..... 1831
49.2.17 MSICAPLST—Offset 8Ch ..... 1831
49.2.18 MSICTL—Offset 8Eh ..... 1832
49.2.19 MSIADDR—Offset 90h ..... 1832
49.2.20 MSIDATA—Offset 98h ..... 1833
49.2.21 MSIMSK—Offset 9Ch ..... 1834
49.2.22 PLKCTL—Offset EAh ..... 1834
49.3 Registers in Memory Space—SMTBAR ..... 1835
49.3.1 GCTRL—Offset 0h ..... 1835
49.3.2 SMTICL—Offset 8h ..... 1836
49.3.3 ERRINTMSK—Offset 10h ..... 1837
49.3.4 ERRAERMSK—Offset 14h ..... 1838
49.3.5 ERRSTS—Offset 18h ..... 1839
49.3.6 ERRINFO—Offset 1Ch ..... 1840
49.3.7 MDBA—Offset 100h ..... 1840
49.3.8 MCTRL—Offset 108h ..... 1841
49.3.9 MSTS—Offset 10Ch ..... 1842
49.3.10 MDS—Offset 110h ..... 1843
49.3.11 RPOLICY—Offset 114h ..... 1844
49.3.12 TBBA—Offset 200h ..... 1846
49.3.13 TCTRL—Offset 208h ..... 1847
49.3.14 TSTS—Offset 20Ch ..... 1848
49.3.15 TBS—Offset 210h ..... 1849
49.3.16 HTHP—Offset 218h ..... 1849
49.3.17 FTTP—Offset 21Ch ..... 1850
49.3.18 TRxCTRL—Offset 220h ..... 1850
49.3.19 TRxSTS—Offset 224h ..... 1851
49.3.20 TACTRL—Offset 228h ..... 1852
49.3.21 TPOLICY—Offset 22Ch ..... 1853
49.3.22 GPBRCTRL—Offset 240h ..... 1855
49.3.23 GPBRDBUF—Offset 244h ..... 1856
49.3.24 SMTARPCTRL—Offset 280h ..... 1857
49.3.25 UDID0—Offset 290h ..... 1858
49.3.26 UUDID0—Offset 298h ..... 1859
49.3.27 UDID1—Offset 2A0h ..... 1860
49.3.28 UUDID1—Offset 2A8h ..... 1861
49.3.29 SPGT—Offset 300h ..... 1862
49.3.30 SPMT—Offset 304h ..... 1863
49.3.31 SPST—Offset 308h ..... 1864
49.3.32 SMBFT—Offset 30Ch ..... 1865
49.3.33 CLTC—Offset 310h ..... 1866
49.3.34 DCLKGT—Offset 380h ..... 1866
49.3.35 SUSCHKB—Offset 384h ..... 1867
49.3.36 DBCTRL—Offset 388h ..... 1868
49.3.37 DBSTS—Offset 38Ch ..... 1869

**50 SATA Controllers - B0, D(19, 20), F0 ..... 1870**
50.1 Introduction and Index ..... 1870
50.1.1 Host Configuration Space ..... 1871
50.1.2 Host Memory Space—MXTBA ..... 1873
50.1.3 Host Memory Space—ABAR ..... 1874
50.2 Registers in Configuration Space ..... 1879
50.2.1 Identifiers (ID)—Offset 0h ..... 1879
50.2.2 Command (CMD)—Offset 4h ..... 1880
50.2.3 Device Status (STS)—Offset 6h ..... 1881
50.2.4 Revision ID (RID)—Offset 8h ..... 1882
50.2.5 Programming Interface (PI)—Offset 9h ..... 1882
50.2.6 Class Code (CC)—Offset Ah ..... 1882
50.2.7 MSI-X Table Base Address (MXTBA)—Offset 10h ..... 1883
50.2.8 SCMDBA—Offset 18h ..... 1884
50.2.9 SCTLBA—Offset 1Ch ..... 1884
50.2.10 AHCI Index Data Pair Base Address (AIDPBA)—Offset 20h ..... 1885
50.2.11 AHCI Base Address (ABAR)—Offset 24h ..... 1886
50.2.12 Capabilities Pointer (CAP)—Offset 34h ..... 1887
50.2.13 Interrupt Information (INTR)—Offset 3Ch ..... 1887
50.2.14 PCI Power Management Capability ID (PID)—Offset 70h ..... 1887
50.2.15 PCI Power Management Capabilities (PC)—Offset 72h ..... 1888
50.2.16 PCI Power Management Control and Status (PMCS)—Offset 74h ..... 1889
50.2.17 Message Signaled Interrupt Identifier (MID)—Offset 80h ..... 1889
50.2.18 Message Signaled Interrupt Message Control (MC)—Offset 82h ..... 1890
50.2.19 Message Signaled Interrupt Message Address (MA)—Offset 84h ..... 1890
50.2.20 Message Signaled Interrupt Message Data (MD)—Offset 88h ..... 1890
50.2.21 Port Mapping Register (MAP)—Offset 90h ..... 1891
50.2.22 Port Control and Status (PCS)—Offset 94h ..... 1893
50.2.23 SATA General Configuration (SATAGC)—Offset 9Ch ..... 1895
50.2.24 SATA Initialization Register Index (SIRI)—Offset A0h ..... 1896
50.2.25 SATA Initialization Register Data (SIRD)—Offset A4h ..... 1896
50.2.26 Serial ATA Capability Register 0 (SATACR0)—Offset A8h ..... 1897
50.2.27 Serial ATA Capability Register 1 (SATACR1)—Offset ACh ..... 1898
50.2.28 FLR Capability ID (FLRCID)—Offset B0h ..... 1898
50.2.29 FLR Control (FLRCTL)—Offset B4h ..... 1899
50.2.30 MSI-X Message Control (MXC)—Offset D2h ..... 1900
50.2.31 MSI-X Table Offset / Table BIR (MXT)—Offset D4h ..... 1900
50.2.32 MSI-X PBA Offset / PBA BIR (MXP)—Offset D8h ..... 1900
50.2.33 BIST FIS Control/Status (BFCS)—Offset E0h ..... 1901
50.2.34 BIST FIS Transmit Data 1 (BFTD1)—Offset E4h ..... 1903
50.2.35 BIST FIS Transmit Data 2 (BFTD2)—Offset E8h ..... 1903
50.2.36 Manufacturing ID (MFID)—Offset F8h ..... 1903
50.3 Registers in Memory Space—MXTBA ..... 1904
50.3.1 MSI-X Table Entries n Message Lower Address (MXTEnMLA) ..... 1904
50.3.2 MSI-X Table Entries n Message Upper Address (MXTEnMUA) ..... 1904
50.3.3 MSI-X Table Entries n Message Data (MXTEnMD) ..... 1905
50.4 Registers in Memory Space—ABAR ..... 1906
50.4.1 HBA Capabilities (GHC_CAP)—Offset 0h ..... 1906
50.4.2 Global HBA Control (GHC)—Offset 4h ..... 1908
50.4.3 Interrupt Status Register (IS)—Offset 8h ..... 1909
50.4.4 Ports Implemented (GHC_PI)—Offset Ch ..... 1910
50.4.5 AHCI Version (VS)—Offset 10h ..... 1911
50.4.6 Enclosure Management Control (EM_CTL)—Offset 20h ..... 1912
50.4.7 HBA Capabilities Extended (GHC_CAP2)—Offset 24h ..... 1914
50.4.8 Vendor Specific (VSP)—Offset A0h ..... 1915

50.4.9 Vendor Specific Capabilities Register (VS_CAP)—Offset A4h ..... 1916
50.4.10 RAID Platform ID (RPID)—Offset C0h ..... 1917
50.4.11 SW Feature Mask (SFM)—Offset C8h ..... 1918
50.4.12 Port [0-7] Command List Base Address (PxCLB0)—Offset 100h ..... 1919
50.4.13 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU0)—Offset 104h ..... 1919
50.4.14 Port [0-7] FIS Base Address (PxFB0)—Offset 108h ..... 1920
50.4.15 Port [0-7] FIS Base Address Upper 32-bits (PxFBU0)—Offset 10Ch ..... 1920
50.4.16 Port [0-7] Interrupt Status (PxIS0)—Offset 110h ..... 1921
50.4.17 Port [0-7] Interrupt Enable (PxIE0)—Offset 114h ..... 1923
50.4.18 Port [0-7] Command (PxCMD0)—Offset 118h ..... 1924
50.4.19 Port [0-7] Serial ATA Status (PxSSTS0)—Offset 128h ..... 1926
50.4.20 Port [0-7] Serial ATA Control (PxSCTL0)—Offset 12Ch ..... 1926
50.4.21 Port [0-7] Serial ATA Error (PxSERR0)—Offset 130h ..... 1927
50.4.22 Port [0-7] Serial ATA Active (PxSACT0)—Offset 134h ..... 1927
50.4.23 Port [0-7] Commands Issued (PxCI0)—Offset 138h ..... 1927
50.4.24 Port [0-7] SNotification (PxSNTF0)—Offset 13Ch ..... 1928
50.4.25 Port [0-7] Command List Base Address (PxCLB1)—Offset 180h ..... 1928
50.4.26 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU1)—Offset 184h ..... 1928
50.4.27 Port [0-7] FIS Base Address (PxFB1)—Offset 188h ..... 1929
50.4.28 Port [0-7] FIS Base Address Upper 32-bits (PxFBU1)—Offset 18Ch ..... 1929
50.4.29 Port [0-7] Interrupt Status (PxIS1)—Offset 190h ..... 1930
50.4.30 Port [0-7] Interrupt Enable (PxIE1)—Offset 194h ..... 1932
50.4.31 Port [0-7] Command (PxCMD1)—Offset 198h ..... 1933
50.4.32 Port [0-7] Serial ATA Status (PxSSTS1)—Offset 1A8h ..... 1935
50.4.33 Port [0-7] Serial ATA Control (PxSCTL1)—Offset 1ACh ..... 1935
50.4.34 Port [0-7] Serial ATA Error (PxSERR1)—Offset 1B0h ..... 1936
50.4.35 Port [0-7] Serial ATA Active (PxSACT1)—Offset 1B4h ..... 1936
50.4.36 Port [0-7] Commands Issued (PxCI1)—Offset 1B8h ..... 1936
50.4.37 Port [0-7] SNotification (PxSNTF1)—Offset 1BCh ..... 1937
50.4.38 Port [0-7] Command List Base Address (PxCLB2)—Offset 200h ..... 1937
50.4.39 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU2)—Offset 204h ..... 1937
50.4.40 Port [0-7] FIS Base Address (PxFB2)—Offset 208h ..... 1938
50.4.41 Port [0-7] FIS Base Address Upper 32-bits (PxFBU2)—Offset 20Ch ..... 1938
50.4.42 Port [0-7] Interrupt Status (PxIS2)—Offset 210h ..... 1939
50.4.43 Port [0-7] Interrupt Enable (PxIE2)—Offset 214h ..... 1941
50.4.44 Port [0-7] Command (PxCMD2)—Offset 218h ..... 1942
50.4.45 Port [0-7] Serial ATA Status (PxSSTS2)—Offset 228h ..... 1944
50.4.46 Port [0-7] Serial ATA Control (PxSCTL2)—Offset 22Ch ..... 1944
50.4.47 Port [0-7] Serial ATA Error (PxSERR2)—Offset 230h ..... 1945
50.4.48 Port [0-7] Serial ATA Active (PxSACT2)—Offset 234h ..... 1945
50.4.49 Port [0-7] Commands Issued (PxCI2)—Offset 238h ..... 1945
50.4.50 Port [0-7] SNotification (PxSNTF2)—Offset 23Ch ..... 1946
50.4.51 Port [0-7] Command List Base Address (PxCLB3)—Offset 280h ..... 1946
50.4.52 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU3)—Offset 284h ..... 1946
50.4.53 Port [0-7] FIS Base Address (PxFB3)—Offset 288h ..... 1947
50.4.54 Port [0-7] FIS Base Address Upper 32-bits (PxFBU3)—Offset 28Ch ..... 1947
50.4.55 Port [0-7] Interrupt Status (PxIS3)—Offset 290h ..... 1948
50.4.56 Port [0-7] Interrupt Enable (PxIE3)—Offset 294h ..... 1950
50.4.57 Port [0-7] Command (PxCMD3)—Offset 298h ..... 1951
50.4.58 Port [0-7] Serial ATA Status (PxSSTS3)—Offset 2A8h ..... 1953
50.4.59 Port [0-7] Serial ATA Control (PxSCTL3)—Offset 2ACh ..... 1953

50.4.60 Port [0-7] Serial ATA Error (PxSERR3)—Offset 2B0h ..... 1954
50.4.61 Port [0-7] Serial ATA Active (PxSACT3)—Offset 2B4h ..... 1954
50.4.62 Port [0-7] Commands Issued (PxCI3)—Offset 2B8h ..... 1954
50.4.63 Port [0-7] SNotification (PxSNTF3)—Offset 2BCh ..... 1955
50.4.64 Port [0-7] Command List Base Address (PxCLB4)—Offset 300h ..... 1955
50.4.65 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU4)—Offset 304h..... 1955
50.4.66 Port [0-7] FIS Base Address (PxFB4)—Offset 308h..... 1956
50.4.67 Port [0-7] FIS Base Address Upper 32-bits (PxFBU4)—Offset 30Ch ..... 1956
50.4.68 Port [0-7] Interrupt Status (PxIS4)—Offset 310h..... 1957
50.4.69 Port [0-7] Interrupt Enable (PxIE4)—Offset 314h..... 1959
50.4.70 Port [0-7] Command (PxCMD4)—Offset 318h..... 1960
50.4.71 Port [0-7] Serial ATA Status (PxSSTS4)—Offset 328h..... 1962
50.4.72 Port [0-7] Serial ATA Control (PxSCTL4)—Offset 32Ch..... 1962
50.4.73 Port [0-7] Serial ATA Error (PxSERR4)—Offset 330h..... 1963
50.4.74 Port [0-7] Serial ATA Active (PxSACT4)—Offset 334h ..... 1963
50.4.75 Port [0-7] Commands Issued (PxCI4)—Offset 338h..... 1963
50.4.76 Port [0-7] SNotification (PxSNTF4)—Offset 33Ch ..... 1964
50.4.77 Port [0-7] Command List Base Address (PxCLB5)—Offset 380h ..... 1964
50.4.78 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU5)—Offset 384h..... 1964
50.4.79 Port [0-7] FIS Base Address (PxFB5)—Offset 388h..... 1965
50.4.80 Port [0-7] FIS Base Address Upper 32-bits (PxFBU5)—Offset 38Ch ..... 1965
50.4.81 Port [0-7] Interrupt Status (PxIS5)—Offset 390h..... 1966
50.4.82 Port [0-7] Interrupt Enable (PxIE5)—Offset 394h..... 1968
50.4.83 Port [0-7] Serial ATA Status (PxSSTS5)—Offset 3A8h..... 1969
50.4.84 Port [0-7] Serial ATA Control (PxSCTL5)—Offset 3ACh..... 1969
50.4.85 Port [0-7] Serial ATA Error (PxSERR5)—Offset 3B0h ..... 1970
50.4.86 Port [0-7] Serial ATA Active (PxSACT5)—Offset 3B4h ..... 1970
50.4.87 Port [0-7] Commands Issued (PxCI5)—Offset 3B8h ..... 1970
50.4.88 Port [0-7] SNotification (PxSNTF5)—Offset 3BCh ..... 1971
50.4.89 Port [0-7] Device Sleep (PxDEVSLP5)—Offset 3C4h ..... 1972
50.4.90 Port [0-7] Command List Base Address (PxCLB6)—Offset 400h ..... 1972
50.4.91 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU6)—Offset 404h..... 1973
50.4.92 Port [0-7] FIS Base Address (PxFB6)—Offset 408h..... 1973
50.4.93 Port [0-7] FIS Base Address Upper 32-bits (PxFBU6)—Offset 40Ch ..... 1973
50.4.94 Port [0-7] Interrupt Status (PxIS6)—Offset 410h..... 1974
50.4.95 Port [0-7] Interrupt Enable (PxIE6)—Offset 414h..... 1976
50.4.96 Port [0-7] Command (PxCMD6)—Offset 418h..... 1977
50.4.97 Port [0-7] Task File Data (PxTFD6)—Offset 420h ..... 1979
50.4.98 Port [0-7] Signature (PxSIG6)—Offset 424h..... 1979
50.4.99 Port [0-7] Serial ATA Status (PxSSTS6)—Offset 428h..... 1980
50.4.100 Port [0-7] Serial ATA Control (PxSCTL6)—Offset 42Ch..... 1980
50.4.101 Port [0-7] Serial ATA Error (PxSERR6)—Offset 430h..... 1981
50.4.102 Port [0-7] Serial ATA Active (PxSACT6)—Offset 434h ..... 1981
50.4.103 Port [0-7] Commands Issued (PxCI6)—Offset 438h..... 1981
50.4.104 Port [0-7] SNotification (PxSNTF6)—Offset 43Ch ..... 1982
50.4.105 Port [0-7] Device Sleep (PxDEVSLP6)—Offset 444h..... 1982
50.4.106 Port [0-7] Command List Base Address (PxCLB7)—Offset 480h ..... 1983
50.4.107 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU7)—Offset 484h..... 1983
50.4.108 Port [0-7] FIS Base Address (PxFB7)—Offset 488h..... 1984
50.4.109 Port [0-7] FIS Base Address Upper 32-bits (PxFBU7)—Offset 48Ch ..... 1984
50.4.110 Port [0-7] Interrupt Status (PxIS7)—Offset 490h..... 1985

50.4.111 Port [0-7] Interrupt Enable (PxIE7)—Offset 494h ..... 1987
50.4.112 Port [0-7] Command (PxCMD7)—Offset 498h ..... 1988
50.4.113 Port [0-7] Task File Data (PxTFD7)—Offset 4A0h ..... 1990
50.4.114 Port [0-7] Signature (PxSIG7)—Offset 4A4h ..... 1990
50.4.115 Port [0-7] Serial ATA Status (PxSSTS7)—Offset 4A8h ..... 1991
50.4.116 Port [0-7] Serial ATA Control (PxSCTL7)—Offset 4ACh ..... 1991
50.4.117 Port [0-7] Serial ATA Error (PxSERR7)—Offset 4B0h ..... 1992
50.4.118 Port [0-7] Serial ATA Active (PxSACT7)—Offset 4B4h ..... 1992
50.4.119 Port [0-7] Commands Issued (PxCI7)—Offset 4B8h ..... 1992
50.4.120 Port [0-7] SNotification (PxSNTF7)—Offset 4BCh ..... 1993
50.4.121 Enclosure Management Message Format (EM_MF)—Offset 580h ..... 1993
50.4.122 Enclosure Management LED (EM_LED)—Offset 584h ..... 1994

**51 USB Combo Controller - B0, D21, F0 ..... 1995**
51.1 Introduction and Index ..... 1995
51.1.1 Host Configuration Space ..... 1996
51.1.2 Host Memory Space—MBAR ..... 1998
51.2 Registers in Configuration Space ..... 2006
51.2.1 Vendor ID (VID)—Offset 0h ..... 2006
51.2.2 Device ID (DID)—Offset 2h ..... 2006
51.2.3 Command (CMD)—Offset 4h ..... 2007
51.2.4 Device Status (STS)—Offset 6h ..... 2008
51.2.5 Revision ID (RID)—Offset 8h ..... 2010
51.2.6 Programming Interface (PI)—Offset 9h ..... 2010
51.2.7 Sub Class Code (SCC)—Offset Ah ..... 2010
51.2.8 Base Class Code (BCC)—Offset Bh ..... 2010
51.2.9 Master Latency Timer (MLT)—Offset Dh ..... 2011
51.2.10 Header Type (HT)—Offset Eh ..... 2011
51.2.11 Memory Base Address (MBAR)—Offset 10h ..... 2012
51.2.12 USB Subsystem Vendor ID (SSVID)—Offset 2Ch ..... 2012
51.2.13 USB Subsystem ID (SSID)—Offset 2Eh ..... 2013
51.2.14 Capabilities Pointer (CAP_PTR)—Offset 34h ..... 2013
51.2.15 Interrupt Line (ILINE)—Offset 3Ch ..... 2013
51.2.16 Interrupt Pin (IPIN)—Offset 3Dh ..... 2014
51.2.17 XHC System Bus Configuration 1 (XHCC1)—Offset 40h ..... 2014
51.2.18 XHC System Bus Configuration 2 (XHCC2)—Offset 44h ..... 2016
51.2.19 Clock Gating (XHCLKGTEN)—Offset 50h ..... 2018
51.2.20 Audio Time Synchronization (AUDSYNC)—Offset 58h ..... 2021
51.2.21 Serial Bus Release Number (SBRN)—Offset 60h ..... 2021
51.2.22 Frame Length Adjustment (FLADJ)—Offset 61h ..... 2022
51.2.23 Best Effort Service Latency (BESL)—Offset 62h ..... 2023
51.2.24 PCI Power Management Capability ID (PM_CID)—Offset 70h ..... 2023
51.2.25 Next Item Pointer #1 (PM_NEXT)—Offset 71h ..... 2023
51.2.26 Power Management Capabilities (PM_CAP)—Offset 72h ..... 2024
51.2.27 Power Management Control/Status (PM_CS)—Offset 74h ..... 2025
51.2.28 Message Signaled Interrupt CID (MSI_CID)—Offset 80h ..... 2026
51.2.29 Next item pointer (MSI_NEXT)—Offset 81h ..... 2026
51.2.30 Message Signaled Interrupt Message Control (MSI_MCTL)—Offset 82h ..... 2027
51.2.31 Message Signaled Interrupt Message Address (MSI_MAD)—Offset 84h ..... 2028
51.2.32 Message Signaled Interrupt Upper Address (MSI_MUAD)—Offset 88h ..... 2028
51.2.33 Message Signaled Interrupt Message Data (MSI_MD)—Offset 8Ch ..... 2028
51.2.34 Device Idle Capability (DEVIDLE)—Offset 90h ..... 2029
51.2.35 Vendor Specific Header (VSHDR)—Offset 94h ..... 2029
51.2.36 SW LTR POINTER (SWLTRPTR)—Offset 98h ..... 2030
51.2.37 Device Idle Pointer Register (DEVIDLEPTR)—Offset 9Ch ..... 2030

51.2.38 Device Idle Power ON Latency (DEVIDLEPOL)—Offset A0h ..... 2031
51.2.39 High Speed Configuration 2 (HSCFG2)—Offset A4h ..... 2032
51.2.40 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1)—Offset B0h ..... 2033
51.2.41 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1)—Offset D0h ..... 2033
51.2.42 XHCC3—Offset FCh ..... 2034
51.3 Registers in Memory Space—MBAR ..... 2035
51.3.1 Capability Registers Length (CAPLENGTH)—Offset 0h ..... 2035
51.3.2 Host Controller Interface Version Number (HCIVERSION)—Offset 2h ..... 2035
51.3.3 Structural Parameters 1 (HCSPARAMS1)—Offset 4h ..... 2036
51.3.4 Structural Parameters 2 (HCSPARAMS2)—Offset 8h ..... 2036
51.3.5 Structural Parameters 3 (HCSPARAMS3)—Offset Ch ..... 2037
51.3.6 Capability Parameters (HCCPARAMS)—Offset 10h ..... 2038
51.3.7 Doorbell Offset (DBOFF)—Offset 14h ..... 2039
51.3.8 Runtime Register Space Offset (RTSOFF)—Offset 18h ..... 2039
51.3.9 USB Command (USBCMD)—Offset 80h ..... 2040
51.3.10 USB Status (USBSTS)—Offset 84h ..... 2041
51.3.11 Page Size (PAGESIZE)—Offset 88h ..... 2042
51.3.12 Device Notification Control (DNCTRL)—Offset 94h ..... 2042
51.3.13 Command Ring Low (CRCR_LO)—Offset 98h ..... 2043
51.3.14 Command Ring High (CRCR_HI)—Offset 9Ch ..... 2043
51.3.15 Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h ..... 2044
51.3.16 Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h ..... 2044
51.3.17 Configure (CONFIG)—Offset B8h ..... 2044
51.3.18 Port Status and Control USB2 (PORTSC1)—Offset 480h ..... 2045
51.3.19 Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h ..... 2047
51.3.20 Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch ..... 2048
51.3.21 Port Status and Control USB2 (PORTSC2)—Offset 490h ..... 2049
51.3.22 Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h ..... 2051
51.3.23 Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch ..... 2052
51.3.24 Port Status and Control USB2 (PORTSC3)—Offset 4A0h ..... 2053
51.3.25 Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h ..... 2055
51.3.26 Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh ..... 2056
51.3.27 Port Status and Control USB2 (PORTSC4)—Offset 4B0h ..... 2057
51.3.28 Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h ..... 2059
51.3.29 Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh ..... 2060
51.3.30 Port Status and Control USB3 (PORTSC5)—Offset 4C0h ..... 2061
51.3.31 Port Power Management Status and Control USB3 (PORTPMSC5)—Offset 4C4h ..... 2064
51.3.32 USB3 Port Link Info (PORTLI5)—Offset 4C8h ..... 2064
51.3.33 Port Status and Control USB3 (PORTSC6)—Offset 4D0h ..... 2065
51.3.34 Port Power Management Status and Control USB3 (PORTPMSC6)—Offset 4D4h ..... 2068
51.3.35 USB3 Port Link Info (PORTLI6)—Offset 4D8h ..... 2068
51.3.36 Port Status and Control USB3 (PORTSC7)—Offset 4E0h ..... 2069
51.3.37 Port Power Management Status and Control USB3 (PORTPMSC7)—Offset 4E4h ..... 2072
51.3.38 USB3 Port Link Info (PORTLI7)—Offset 4E8h ..... 2072
51.3.39 Port Status and Control USB3 (PORTSC8)—Offset 4F0h ..... 2073
51.3.40 Port Power Management Status and Control USB3 (PORTPMSC8)—Offset 4F4h ..... 2076

51.3.41 USB3 Port Link Info (PORTLI8)—Offset 4F8h....2076
51.3.42 Microframe Index (RTMFINDEX)—Offset 2000h....2076
51.3.43 Interrupter 1 Management (IMAN0)—Offset 2020h....2077
51.3.44 Interrupter 1 Moderation (IMOD0)—Offset 2024h....2077
51.3.45 Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h....2077
51.3.46 Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h....2078
51.3.47 Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h....2078
51.3.48 Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h....2079
51.3.49 Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch....2079
51.3.50 Interrupter 2 Management (IMAN1)—Offset 2040h....2080
51.3.51 Interrupter 2 Moderation (IMOD1)—Offset 2044h....2080
51.3.52 Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h....2080
51.3.53 Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h....2081
51.3.54 Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h....2081
51.3.55 Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h....2081
51.3.56 Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch....2082
51.3.57 Interrupter 3 Management (IMAN2)—Offset 2060h....2082
51.3.58 Interrupter 3 Moderation (IMOD2)—Offset 2064h....2082
51.3.59 Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h....2083
51.3.60 Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h....2083
51.3.61 Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h....2083
51.3.62 Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h....2084
51.3.63 Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch....2084
51.3.64 Interrupter 4 Management (IMAN3)—Offset 2080h....2084
51.3.65 Interrupter 4 Moderation (IMOD3)—Offset 2084h....2085
51.3.66 Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h....2085
51.3.67 Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h....2085
51.3.68 Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h....2086
51.3.69 Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h....2086
51.3.70 Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch....2086
51.3.71 Interrupter 5 Management (IMAN4)—Offset 20A0h....2087
51.3.72 Interrupter 5 Moderation (IMOD4)—Offset 20A4h....2087
51.3.73 Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h....2087
51.3.74 Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h....2088
51.3.75 Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h....2088
51.3.76 Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h....2088
51.3.77 Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh....2089
51.3.78 Interrupter 6 Management (IMAN5)—Offset 20C0h....2089
51.3.79 Interrupter 6 Moderation (IMOD5)—Offset 20C4h....2089
51.3.80 Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h....2090
51.3.81 Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h....2090
51.3.82 Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h....2090
51.3.83 Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h....2091
51.3.84 Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh....2091

51.3.85 Interrupter 7 Management (IMAN6)—Offset 20E0h ..... 2091
51.3.86 Interrupter 7 Moderation (IMOD6)—Offset 20E4h ..... 2092
51.3.87 Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h ..... 2092
51.3.88 Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h ..... 2092
51.3.89 Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h ..... 2093
51.3.90 Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h ..... 2093
51.3.91 Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh ..... 2093
51.3.92 Interrupter 8 Management (IMAN7)—Offset 2100h ..... 2094
51.3.93 Interrupter 8 Moderation (IMOD7)—Offset 2104h ..... 2094
51.3.94 Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h ..... 2094
51.3.95 Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h ..... 2095
51.3.96 Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h ..... 2095
51.3.97 Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h ..... 2095
51.3.98 Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch ..... 2096
51.3.99 Door Bell 1 (DB0)—Offset 3000h ..... 2096
51.3.100 Door Bell 2 (DB1)—Offset 3004h ..... 2097
51.3.101 Door Bell 3 (DB2)—Offset 3008h ..... 2097
51.3.102 Door Bell 4 (DB3)—Offset 300Ch ..... 2098
51.3.103 Door Bell 5 (DB4)—Offset 3010h ..... 2098
51.3.104 Door Bell 6 (DB5)—Offset 3014h ..... 2099
51.3.105 Door Bell 7 (DB6)—Offset 3018h ..... 2099
51.3.106 Door Bell 8 (DB7)—Offset 301Ch ..... 2100
51.3.107 Door Bell 9 (DB8)—Offset 3020h ..... 2100
51.3.108 Door Bell 10 (DB9)—Offset 3024h ..... 2101
51.3.109 Door Bell 11 (DB10)—Offset 3028h ..... 2101
51.3.110 Door Bell 12 (DB11)—Offset 302Ch ..... 2102
51.3.111 Door Bell 13 (DB12)—Offset 3030h ..... 2102
51.3.112 Door Bell 14 (DB13)—Offset 3034h ..... 2103
51.3.113 Door Bell 15 (DB14)—Offset 3038h ..... 2103
51.3.114 Door Bell 16 (DB15)—Offset 303Ch ..... 2104
51.3.115 Door Bell 17 (DB16)—Offset 3040h ..... 2104
51.3.116 Door Bell 18 (DB17)—Offset 3044h ..... 2105
51.3.117 Door Bell 19 (DB18)—Offset 3048h ..... 2105
51.3.118 Door Bell 20 (DB19)—Offset 304Ch ..... 2106
51.3.119 Door Bell 21 (DB20)—Offset 3050h ..... 2106
51.3.120 Door Bell 22 (DB21)—Offset 3054h ..... 2107
51.3.121 Door Bell 23 (DB22)—Offset 3058h ..... 2107
51.3.122 Door Bell 24 (DB23)—Offset 305Ch ..... 2108
51.3.123 Door Bell 25 (DB24)—Offset 3060h ..... 2108
51.3.124 Door Bell 26 (DB25)—Offset 3064h ..... 2109
51.3.125 Door Bell 27 (DB26)—Offset 3068h ..... 2109
51.3.126 Door Bell 28 (DB27)—Offset 306Ch ..... 2110
51.3.127 Door Bell 29 (DB28)—Offset 3070h ..... 2110
51.3.128 Door Bell 30 (DB29)—Offset 3074h ..... 2111
51.3.129 Door Bell 31 (DB30)—Offset 3078h ..... 2111
51.3.130 Door Bell 32 (DB31)—Offset 307Ch ..... 2112
51.3.131 Door Bell 32 (DB32)—Offset 3080h ..... 2112
51.3.132 XECP_SUPP_USB2_1—Offset 8004h ..... 2113
51.3.133 XECP_SUPP_USB2_2—Offset 8008h ..... 2113
51.3.134 XECP_SUPP_USB3_1—Offset 8024h ..... 2114
51.3.135 XECP_SUPP_USB3_2—Offset 8028h ..... 2114

51.3.136 Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h .....2114
51.3.137 Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h .....2115
51.3.138 Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch .....2115
51.3.139 Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h .....2116
51.3.140 Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h .....2117
51.3.141 Power Management Control (PMCTRL_REG)—Offset 80A4h .....2118
51.3.142 PGCB Control (PGCBCTRL_REG)—Offset 80A8h .....2120
51.3.143 D0I3 Control (DOI3CTRL_REG)—Offset 80ACh .....2123
51.3.144 HOST_CTRL_MISC_REG—Offset 80B0h .....2124
51.3.145 reg_SSPITPE_type (SSPITPE)—Offset 80BCh .....2126
51.3.146 AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h .....2127
51.3.147 Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h .....2130
51.3.148 High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h .....2131
51.3.149 Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh.....2131
51.3.150 System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h....2132
51.3.151 Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h .....2133
51.3.152 AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h .....2134
51.3.153 Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h .....2137
51.3.154 SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh .....2138
51.3.155 USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h .....2140
51.3.156 USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h .....2142
51.3.157 USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h .....2142
51.3.158 USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh .....2143
51.3.159 Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch ...2143
51.3.160 USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h .....2144
51.3.161 Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h .....2144
51.3.162 AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h .....2146
51.3.163 USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h .....2148
51.3.164 USB Power Gating Control (USB_PGC)—Offset 8168h .....2149
51.3.165 xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch .....2150
51.3.166 USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h .....2152
51.3.167 xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h .....2153
51.3.168 xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h .....2154
51.3.169 xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch .....2154
51.3.170 xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h .....2155
51.3.171 xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h .....2155
51.3.172 HOST_CTRL_BW_MAX3_REG—Offset 8188h .....2156
51.3.173 THRM_HOST_CTRL_REG—Offset 819Ch .....2157
51.3.174 THRM_HOST_CTRL_REG2—Offset 81B4h .....2158

51.3.175 LFPSONCOUNT_REG—Offset 81B8h ..... 2158
51.3.176 reg_D0i2_CTRL_REG_type (D0i2_CTRL_REG)—Offset 81BCh ..... 2160
51.3.177 reg_D0i2_SCH_ALARM_CTRL_REG_type (D0i2_SCH_ALARM_CTRL_REG)—Offset 81C0h ..... 2162
51.3.178 reg_USB2PMCTRL_REG_type (USB2PMCTRL_REG)—Offset 81C4h ..... 2163
51.3.179 ECC_PARITY_ERROR_LOG_REG—Offset 83F8h ..... 2164
51.3.180 ECC_POISONING_CTRL_REG—Offset 83FCh ..... 2166
51.3.181 USB2_PORT_STATE_REG—Offset 8400h ..... 2167
51.3.182 USB2_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h ..... 2167
51.3.183 FUS1_REG—Offset 8410h ..... 2168
51.3.184 FUS2_REG—Offset 8414h ..... 2169
51.3.185 STRAP1_REG—Offset 841Ch ..... 2169
51.3.186 STRAP2_REG—Offset 8420h ..... 2169
51.3.187 STRAP3_REG—Offset 8424h ..... 2170
51.3.188 DFT_REG1—Offset 8430h ..... 2171
51.3.189 DFT_REG2—Offset 8434h ..... 2172
51.3.190 DFT_REG3—Offset 8438h ..... 2173
51.3.191 dft_reg4 (DFT_REG4)—Offset 843Ch ..... 2173
51.3.192 dft_reg5 (DFT_REG5)—Offset 8440h ..... 2174
51.3.193 XECP_CMDM_STS0—Offset 8448h ..... 2175
51.3.194 XECP_CMDM_STS2—Offset 8450h ..... 2177
51.3.195 XECP_CMDM_STS3—Offset 8454h ..... 2177
51.3.196 AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h ..... 2177
51.3.197 Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h ..... 2178
51.3.198 USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch ..... 2179
51.3.199 USB2 Port Disable Override (USB2PDO)—Offset 84F8h ..... 2180
51.3.200 USB3 Port Disable Override (USB3PDO)—Offset 84FCh ..... 2180
51.3.201 Command (CMD_MMIO)—Offset 8604h ..... 2181
51.3.202 Device Status (STS_MMIO)—Offset 8606h ..... 2182
51.3.203 Revision ID (RID_MMIO)—Offset 8608h ..... 2184
51.3.204 Programming Interface (PI_MMIO)—Offset 8609h ..... 2184
51.3.205 Sub Class Code (SCC_MMIO)—Offset 860Ah ..... 2184
51.3.206 Base Class Code (BCC_MMIO)—Offset 860Bh ..... 2185
51.3.207 Master Latency Timer (MLT_MMIO)—Offset 860Dh ..... 2185
51.3.208 Header Type (HT_MMIO)—Offset 860Eh ..... 2185
51.3.209 Memory Base Address (MBAR_MMIO)—Offset 8610h ..... 2186
51.3.210 USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch ..... 2186
51.3.211 USB Subsystem ID (SSID_MMIO)—Offset 862Eh ..... 2187
51.3.212 Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h ..... 2187
51.3.213 Interrupt Line (ILINE_MMIO)—Offset 863Ch ..... 2187
51.3.214 Interrupt Pin (IPIN_MMIO)—Offset 863Dh ..... 2188
51.3.215 XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h ..... 2189
51.3.216 XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h ..... 2191
51.3.217 Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h ..... 2193
51.3.218 Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h ..... 2196
51.3.219 Serial Bus Release Number (SBRN_MMIO)—Offset 8660h ..... 2196
51.3.220 Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h ..... 2197
51.3.221 Best Effort Service Latency (BESL_MMIO)—Offset 8662h ..... 2197
51.3.222 PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h ..... 2198
51.3.223 Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h ..... 2198
51.3.224 Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h ..... 2199
51.3.225 Power Management Control/Status (PM_CS_MMIO)—Offset 8674h ..... 2200
51.3.226 Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h ..... 2201
51.3.227 Next item pointer (MSI_NEXT_MMIO)—Offset 8681h ..... 2201

51.3.228 Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h ....2202
51.3.229 Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h ....2202
51.3.230 Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h ....2203
51.3.231 Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch ....2203
51.3.232 Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h ....2204
51.3.233 Vendor Specific Header (VSHDR_MMIO)—Offset 8694h ....2204
51.3.234 SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h ....2205
51.3.235 Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch ....2205
51.3.236 Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h ....2206
51.3.237 High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h ....2207
51.3.238 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h ....2208
51.3.239 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h ....2208
51.3.240 XHCC3 (XHCC3_MMIO)—Offset 86FCh ....2209
51.3.241 Debug Capability ID Register (DCID)—Offset 8700h ....2209
51.3.242 Debug Capability Doorbell Register (DCDB)—Offset 8704h ....2210
51.3.243 Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h ....2210
51.3.244 Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h ....2211
51.3.245 Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h ....2211
51.3.246 Debug Capability Control Register (DCCTRL)—Offset 8720h ....2212
51.3.247 Debug Capability Status Register (DCST)—Offset 8724h ....2213
51.3.248 Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h ....2214
51.3.249 Debug Capability Context Pointer Register (DCCP)—Offset 8730h ....2216
51.3.250 Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h ....2216
51.3.251 Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch ....2217
51.3.252 Debug Capability Descriptor Parameters (DCDP)—Offset 8740h ....2217
51.3.253 Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h ....2218
51.3.254 DBC Control Register 1 (DBCCTL_REG)—Offset 8760h ....2220

**52 System Management - B0, D24, F(0, 1, 3, 4) ....2221**

52.1 Introduction and Index ....2221
52.1.1 Host Configuration Space—HECI1 (F:0) ....2222
52.1.2 Host Configuration Space—HECI2 (F:1) ....2224
52.1.3 Host Configuration Space—HECI3 (F:4) ....2225
52.1.4 Host Configuration Space—KT (F:3) ....2226
52.1.5 Host Configuration Space—IDE (F:2) ....2227
52.1.6 Host Memory Space—HECI1_MMIO_MBAR ....2228
52.1.7 Host Memory Space—HECI2_MMIO_MBAR ....2229
52.1.8 Host Memory Space—HECI3_MMIO_MBAR ....2230
52.1.9 Host Memory Space—KT_HOST_MEMBAR ....2231
52.1.10 Host Memory Space—Fixed Address ....2232
52.1.11 Host I/O Space—KT_HOST_IOBAR ....2233
52.1.12 Host I/O Space—IDE_HOST_PCMDIOBAR ....2234
52.1.13 Host I/O Space—IDE_HOST_PCTLIOBAR ....2235

52.1.14 Host I/O Space—IDE_HOST_BMIOBAR ..... 2236
52.1.15 Sideband Registers ..... 2237
52.2 Registers in Configuration Space—HECI1 ..... 2239
52.2.1 Identifiers (HECI1_ID)—Offset 0h ..... 2239
52.2.2 Command (HECI1_CMD)—Offset 4h ..... 2240
52.2.3 Status (HECI1_STS)—Offset 6h ..... 2241
52.2.4 Revision ID and Class Code (HECI1_RID_CC)—Offset 8h ..... 2242
52.2.5 Cache Line Size (HECI1_CLS)—Offset Ch ..... 2242
52.2.6 Master Latency Timer (HECI1_MLT)—Offset Dh ..... 2242
52.2.7 Header Type (HECI1_HTYPE)—Offset Eh ..... 2243
52.2.8 Built In Self-Test (HECI1_BIST)—Offset Fh ..... 2243
52.2.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h ..... 2243
52.2.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h ..... 2244
52.2.11 Sub System Identifiers (HECI1_SS)—Offset 2Ch ..... 2244
52.2.12 Capabilities Pointer (HECI1_CAP)—Offset 34h ..... 2244
52.2.13 Interrupt Information (HECI1_INTR)—Offset 3Ch ..... 2245
52.2.14 Minimum Grant (HECI1_MGNT)—Offset 3Eh ..... 2245
52.2.15 Maximum Latency (HECI1_MLAT)—Offset 3Fh ..... 2245
52.2.16 Host Firmware Status (HECI1_HFS)—Offset 40h ..... 2246
52.2.17 Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h ..... 2248
52.2.18 General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h ..... 2249
52.2.19 Host General Status (HECI1_H_GS1)—Offset 4Ch ..... 2250
52.2.20 PCI Power Management Capability ID (HECI1_PID)—Offset 50h ..... 2250
52.2.21 PCI Power Management Capabilities (HECI1_PC)—Offset 52h ..... 2250
52.2.22 PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h ..... 2251
52.2.23 General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h ..... 2251
52.2.24 General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h ..... 2252
52.2.25 General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h ..... 2252
52.2.26 General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch ..... 2252
52.2.27 Host General Status 2 (HECI1_H_GS2)—Offset 70h ..... 2252
52.2.28 Host General Status 3 (HECI1_H_GS3)—Offset 74h ..... 2253
52.2.29 Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch ..... 2253
52.2.30 Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh ..... 2253
52.2.31 Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h ..... 2254
52.2.32 Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h ..... 2254
52.2.33 Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h ..... 2254
52.2.34 HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h ..... 2255
52.2.35 Host Extend Register Status (HECI1_HERS)—Offset BCh ..... 2255
52.2.36 Host Extend Register DW1. (HECI1_HER1)—Offset C0h ..... 2256
52.2.37 Host Extend Register DW2. (HECI1_HER2)—Offset C4h ..... 2256
52.2.38 Host Extend Register DW3. (HECI1_HER3)—Offset C8h ..... 2256
52.2.39 Host Extend Register DW4. (HECI1_HER4)—Offset CCh ..... 2256
52.2.40 Host Extend Register DW5. (HECI1_HER5)—Offset D0h ..... 2257
52.2.41 Host Extend Register DW6. (HECI1_HER6)—Offset D4h ..... 2257
52.2.42 Host Extend Register DW7. (HECI1_HER7)—Offset D8h ..... 2257
52.2.43 Host Extend Register DW8. (HECI1_HER8)—Offset DCh ..... 2257
52.2.44 Manufacturer's ID (HECI1_MANID)—Offset F8h ..... 2258
52.3 Registers in Configuration Space—HECI2 ..... 2259
52.3.1 Identifiers (HECI2_ID)—Offset 0h ..... 2259
52.3.2 Command (HECI2_CMD)—Offset 4h ..... 2259
52.3.3 Status (HECI2_STS)—Offset 6h ..... 2261
52.3.4 Revision ID and Class Code (HECI2_RID_CC)—Offset 8h ..... 2262
52.3.5 Cache Line Size (HECI2_CLS)—Offset Ch ..... 2262
52.3.6 Master Latency Timer (HECI2_MLT)—Offset Dh ..... 2262

52.3.7 Header Type (HECI2_HTYPE)—Offset Eh ....2263
52.3.8 Built In Self-Test (HECI2_BIST)—Offset Fh ....2263
52.3.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h ....2263
52.3.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h ....2264
52.3.11 Sub System Identifiers (HECI2_SS)—Offset 2Ch ....2264
52.3.12 Capabilities Pointer (HECI2_CAP)—Offset 34h ....2264
52.3.13 Interrupt Information (HECI2_INTR)—Offset 3Ch ....2265
52.3.14 Minimum Grant (HECI2_MGNT)—Offset 3Eh ....2265
52.3.15 Maximum Latency (HECI2_MLAT)—Offset 3Fh ....2265
52.3.16 Host Firmware Status (HECI2_HFS)—Offset 40h ....2266
52.3.17 General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h ....2268
52.3.18 Host General Status (HECI2_H_GS1)—Offset 4Ch ....2269
52.3.19 PCI Power Management Capability ID (HECI2_PID)—Offset 50h ....2269
52.3.20 PCI Power Management Capabilities (HECI2_PC)—Offset 52h ....2270
52.3.21 PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h ....2271
52.3.22 General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h ....2271
52.3.23 General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h ....2272
52.3.24 General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h ....2272
52.3.25 General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch ....2272
52.3.26 Host General Status 2 (HECI2_H_GS2)—Offset 70h ....2272
52.3.27 Host General Status 3 (HECI2_H_GS3)—Offset 74h ....2273
52.3.28 Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch ....2273
52.3.29 Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh ....2273
52.3.30 Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h ....2274
52.3.31 Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h ....2274
52.3.32 Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h ....2274
52.3.33 HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h ....2275
52.3.34 Manufacturer's ID (HECI2_MANID)—Offset F8h ....2275
52.4 Registers in Configuration Space—HECI3 ....2276
52.4.1 Identifiers (HECI3_ID)—Offset 0h ....2276
52.4.2 Command (HECI3_CMD)—Offset 4h ....2277
52.4.3 Status (HECI3_STS)—Offset 6h ....2278
52.4.4 Revision ID and Class Code (HECI3_RID_CC)—Offset 8h ....2279
52.4.5 Cache Line Size (HECI3_CLS)—Offset Ch ....2279
52.4.6 Master Latency Timer (HECI3_MLT)—Offset Dh ....2279
52.4.7 Header Type (HECI3_HTYPE)—Offset Eh ....2280
52.4.8 Built In Self-Test (HECI3_BIST)—Offset Fh ....2280
52.4.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h ....2280
52.4.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h ....2281
52.4.11 Sub System Identifiers (HECI3_SS)—Offset 2Ch ....2281
52.4.12 Capabilities Pointer (HECI3_CAP)—Offset 34h ....2281
52.4.13 Interrupt Information (HECI3_INTR)—Offset 3Ch ....2282
52.4.14 Minimum Grant (HECI3_MGNT)—Offset 3Eh ....2282
52.4.15 Maximum Latency (HECI3_MLAT)—Offset 3Fh ....2282
52.4.16 Host Firmware Status (HECI3_HFS)—Offset 40h ....2283
52.4.17 General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h ....2285
52.4.18 Host General Status (HECI3_H_GS1)—Offset 4Ch ....2286
52.4.19 PCI Power Management Capability ID (HECI3_PID)—Offset 50h ....2286
52.4.20 PCI Power Management Capabilities (HECI3_PC)—Offset 52h ....2286
52.4.21 PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h ....2287
52.4.22 General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h ....2287
52.4.23 General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h ....2288
52.4.24 General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h ....2288

52.4.25 General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch ..... 2288
52.4.26 Host General Status 2 (HECI3_H_GS2)—Offset 70h ..... 2288
52.4.27 Host General Status 3 (HECI3_H_GS3)—Offset 74h ..... 2289
52.4.28 Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch ..... 2289
52.4.29 Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh .... 2289
52.4.30 Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h ... 2290
52.4.31 Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h ..... 2290
52.4.32 Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h..... 2290
52.4.33 HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h..... 2291
52.4.34 Manufacturer's ID (HECI3_MANID)—Offset F8h ..... 2291
52.5 Registers in Configuration Space—KT (Host)..... 2292
52.5.1 Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h..... 2292
52.5.2 Status and Command (KT_HOST_STS_CMD)—Offset 4h ..... 2293
52.5.3 Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h ..... 2295
52.5.4 BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch ..... 2295
52.5.5 KT IO BAR (KT_HOST_IOBAR)—Offset 10h ..... 2296
52.5.6 KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h ..... 2296
52.5.7 Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h ..... 2297
52.5.8 Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch..... 2297
52.5.9 Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h ..... 2297
52.5.10 Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h ..... 2298
52.5.11 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch ..... 2298
52.5.12 MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h ..... 2299
52.5.13 MSI Message Address (KT_HOST_MSIMA)—Offset 44h ..... 2299
52.5.14 MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h ..... 2300
52.5.15 MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch ..... 2300
52.5.16 Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h ..... 2301
52.5.17 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h ..... 2302
52.5.18 Process Stepping Information (KT_HOST_PSI)—Offset F8h ..... 2303
52.6 Registers in Configuration Space—IDE (Host) ..... 2304
52.6.1 Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h ..... 2304
52.6.2 Status and Command (IDE_HOST_STS_CMD)—Offset 4h..... 2305
52.6.3 Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h ..... 2307
52.6.4 BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch..... 2307
52.6.5 IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h ..... 2308
52.6.6 IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h ..... 2308
52.6.7 IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h ..... 2309
52.6.8 IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch ..... 2309
52.6.9 IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h..... 2310
52.6.10 Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h ..... 2310
52.6.11 Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch ..... 2311
52.6.12 Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h ..... 2311
52.6.13 Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h ..... 2311

52.6.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch ....2312
52.6.15 MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h ....2313
52.6.16 MSI Message Address (IDE_HOST_MSIMA)—Offset 44h ....2313
52.6.17 MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h ....2314
52.6.18 MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch ....2314
52.6.19 Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h ....2315
52.6.20 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h ....2316
52.6.21 Process Stepping Information (IDE_HOST_PSI)—Offset F8h....2317
52.7 Registers in Memory Space—HECI1_MMIO_MBAR ....2318
52.7.1 Host CB Write Window (HECI1_H_CB_WW)—Offset 0h....2318
52.7.2 Host Control and Status Register (HECI1_H_CSR)—Offset 4h....2319
52.7.3 CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h ....2320
52.7.4 CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch....2320
52.7.5 D0i3 Control (HECI1_D0I3C)—Offset 800h ....2321
52.8 Registers in Memory Space—HECI2_MMIO_MBAR ....2322
52.8.1 Host CB Write Window (HECI2_H_CB_WW)—Offset 0h....2322
52.8.2 Host Control and Status Register (HECI2_H_CSR)—Offset 4h....2323
52.8.3 CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h ....2324
52.8.4 CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch....2324
52.8.5 D0i3 Control (HECI2_D0I3C)—Offset 800h ....2325
52.9 Registers in Memory Space—HECI3_MMIO_MBAR ....2326
52.9.1 Host CB Write Window (HECI3_H_CB_WW)—Offset 0h....2326
52.9.2 Host Control and Status Register (HECI3_H_CSR)—Offset 4h....2327
52.9.3 CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h ....2328
52.9.4 CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch....2328
52.9.5 D0i3 Control (HECI3_D0I3C)—Offset 800h ....2329
52.10 Registers in Memory Space—KT_HOST_MEMBAR ....2330
52.10.1 KT Receive Buffer Register (MEM_KTRBR)—Offset 0h....2330
52.10.2 KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h ....2330
52.10.3 KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h ....2331
52.10.4 KT Interrupt Enable register (MEM_KTIER)—Offset 1h....2331
52.10.5 KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h....2332
52.10.6 KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h....2333
52.10.7 KT FIFO Control register (MEM_KTFCR)—Offset 2h ....2334
52.10.8 KT Line Control register (MEM_KTLCR)—Offset 3h ....2335
52.10.9 KT Modem Control register (MEM_KTMCR)—Offset 4h ....2336
52.10.10 KT Line Status register (MEM_KTLSR)—Offset 5h ....2337
52.10.11 KT Modem Status register (MEM_KTMSR)—Offset 6h ....2338
52.10.12 KT Scratch register (MEM_KTSCR)—Offset 7h ....2338
52.11 Registers in Memory Space—Fixed Address....2339
52.11.1 FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h ....2339
52.11.2 FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h...2340
52.11.3 FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h ....2340
52.11.4 FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch ....2340

52.11.5 FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h ..... 2341
52.11.6 FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h ..... 2341
52.11.7 FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h ..... 2341
52.11.8 FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h ..... 2342
52.11.9 Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h . 2342
52.11.10 Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h ..... 2342
52.11.11 Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h ..... 2343
52.11.12 Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch ..... 2343
52.11.13 Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h ..... 2343
52.11.14 Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h ..... 2344
52.11.15 Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h ..... 2344
52.11.16 Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch ..... 2344
52.11.17 Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h ..... 2345
52.11.18 Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h ..... 2345
52.11.19 Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h ..... 2345
52.11.20 Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch ..... 2346
52.11.21 FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h ..... 2346
52.11.22 FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h ..... 2346
52.12 Registers in I/O Space—KT_HOST_IOBAR ..... 2347
52.12.1 KT Receive Buffer Register (IO_KTRBR)—Offset 0h ..... 2347
52.12.2 KT Transmitter Holding Register (IO_KTTHR)—Offset 0h ..... 2347
52.12.3 KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h ..... 2348
52.12.4 KT Interrupt Enable register (IO_KTIER)—Offset 1h ..... 2348
52.12.5 KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h ..... 2349
52.12.6 KT Interrupt Identification Register (IO_KTIIR)—Offset 2h ..... 2349
52.12.7 KT FIFO Control register (IO_KTFCR)—Offset 2h ..... 2350
52.12.8 KT Line Control register (IO_KTLCR)—Offset 3h ..... 2351
52.12.9 KT Modem Control register (IO_KTMCR)—Offset 4h ..... 2352
52.12.10 KT Line Status register (IO_KTLSR)—Offset 5h ..... 2353
52.12.11 KT Modem Status register (IO_KTMSR)—Offset 6h ..... 2354
52.12.12 KT Scratch register (IO_KTSCR)—Offset 7h ..... 2355
52.13 Registers in I/O Space—IDE_HOST_PCMDIOBAR ..... 2356
52.13.1 IDE Data Register (IDEDATA)—Offset 0h ..... 2356
52.13.2 IDE Features Register (IDEFR)—Offset 1h ..... 2356
52.13.3 IDE Error Register DEV0 (IDEERD0)—Offset 1h ..... 2357
52.13.4 IDE Error Register DEV1 (IDEERD1)—Offset 1h ..... 2357
52.13.5 IDE Sector Count In Register (IDESCIR)—Offset 2h ..... 2358
52.13.6 IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h ..... 2358
52.13.7 IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h ..... 2358

52.13.8 IDE Sector Number In Register (IDESNIR)—Offset 3h ..... 2359
52.13.9 IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h ..... 2359
52.13.10 IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h ..... 2360
52.13.11 IDE Cylinder Low In Register (IDECLIR)—Offset 4h ..... 2360
52.13.12 IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h ..... 2361
52.13.13 IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h ..... 2361
52.13.14 IDE Cylinder High In Register (IDECHIR)—Offset 5h ..... 2362
52.13.15 IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h ..... 2362
52.13.16 IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h ..... 2363
52.13.17 IDE Drive/Head In Register (IDEDHIR)—Offset 6h ..... 2363
52.13.18 IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h ..... 2364
52.13.19 IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h ..... 2364
52.13.20 IDE Command Register (IDECR)—Offset 7h ..... 2365
52.13.21 IDE Status Register DEV0 (IDESR0)—Offset 7h ..... 2366
52.13.22 IDE Status Register DEV1 (IDESR1)—Offset 7h ..... 2367
52.14 Registers in I/O Space—IDE_HOST_PCTLIOBAR ..... 2368
52.14.1 IDE Device Control Register (IDDCR)—Offset 2h ..... 2368
52.14.2 IDE Alternate Status Register (IDASR)—Offset 2h ..... 2368
52.15 Registers in I/O Space—IDE_HOST_BMIOBAR ..... 2369
52.15.1 IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h ..... 2369
52.15.2 IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h ..... 2370
52.15.3 IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h ..... 2371
52.15.4 IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h ..... 2373
52.15.5 IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h ..... 2373
52.15.6 IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h ..... 2373
52.15.7 IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h ..... 2374
52.15.8 IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h ..... 2374
52.15.9 IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h ..... 2375
52.15.10 IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h ..... 2375
52.15.11 IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah ..... 2376
52.15.12 IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh ..... 2377
52.15.13 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch ..... 2377
52.15.14 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh ..... 2378
52.15.15 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh ..... 2378
52.15.16 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh ..... 2379
52.16 Sideband Registers—KT (Host) ..... 2380
52.16.1 KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h ..... 2380
52.16.2 BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h ..... 2380
52.17 Sideband Registers—IDE (Host) ..... 2381
52.17.1 IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h ..... 2381
52.17.2 BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h ..... 2381

**53 HSUART Controller - B0, D26, F(0, 1, 2) ..... 2382**
53.1 Introduction and Index ..... 2382
53.1.1 Host Configuration Space ..... 2383
53.1.2 Host Memory Space—MEMBA ..... 2384
53.1.3 Host I/O Space—IOBA ..... 2386
53.2 Registers in Configuration Space ..... 2387
53.2.1 VENDOR_DEVICE_ID—Offset 0h ..... 2387
53.2.2 CMD_STAT_REG—Offset 4h ..... 2388
53.2.3 REV_CLASS_CODE—Offset 8h ..... 2390
53.2.4 CACHE—Offset Ch ..... 2390
53.2.5 IOBA—Offset 10h ..... 2391
53.2.6 MEMBA—Offset 14h ..... 2391
53.2.7 SUBSYSTEM_VENDOR_ID—Offset 2Ch ..... 2392
53.2.8 CAP_PTR—Offset 34h ..... 2392
53.2.9 INTERRUPT—Offset 3Ch ..... 2392
53.2.10 MSIC—Offset 40h ..... 2393
53.2.11 MSIA—Offset 44h ..... 2394
53.2.12 MSID—Offset 48h ..... 2394
53.2.13 PMC—Offset 50h ..... 2395
53.2.14 PMCSR—Offset 54h ..... 2396
53.2.15 DSC—Offset 60h ..... 2397
53.2.16 FUNC_RDCFG_HIDE—Offset 74h ..... 2398
53.3 Registers in Memory Space—MEMBA ..... 2399
53.3.1 RBR_THR_DLL—Offset 0h ..... 2399
53.3.2 IER_DLH—Offset 1h ..... 2399
53.3.3 IIR_FCR—Offset 2h ..... 2400
53.3.4 LCR—Offset 3h ..... 2400
53.3.5 MCR—Offset 4h ..... 2402
53.3.6 LSR—Offset 5h ..... 2404
53.3.7 MSR—Offset 6h ..... 2406
53.3.8 SPR—Offset 7h ..... 2408
53.3.9 UART_FISR—Offset 8h ..... 2408
53.3.10 RFOR—Offset 20h ..... 2408
53.3.11 ABR—Offset 24h ..... 2409
53.3.12 ACR—Offset 28h ..... 2410
53.3.13 PSR—Offset 30h ..... 2410
53.3.14 UCMR—Offset 34h ..... 2410
53.3.15 UCDR—Offset 38h ..... 2411
53.3.16 CH0SR—Offset 80h ..... 2412
53.3.17 CH0CR—Offset 84h ..... 2412
53.3.18 CH0DCR—Offset 88h ..... 2414
53.3.19 CH0BSR—Offset 90h ..... 2415
53.3.20 CH0MTSR—Offset 94h ..... 2415
53.3.21 CH0D0SAR—Offset A0h ..... 2415
53.3.22 CH0D0TSR—Offset A4h ..... 2416
53.3.23 CH0D1SAR—Offset A8h ..... 2416
53.3.24 CH0D1TSR—Offset ACh ..... 2416
53.3.25 CH0D2SAR—Offset B0h ..... 2417
53.3.26 CH0D2TSR—Offset B4h ..... 2417
53.3.27 CH0D3SAR—Offset B8h ..... 2418
53.3.28 CH0D3TSR—Offset BCh ..... 2418
53.3.29 CH1SR—Offset C0h ..... 2419
53.3.30 CH1CR—Offset C4h ..... 2420
53.3.31 CH1DCR—Offset C8h ..... 2421
53.3.32 CH1BSR—Offset D0h ..... 2422

53.3.33 CH1MTSR—Offset D4h ..... 2422
53.3.34 CH1D0SAR—Offset E0h ..... 2422
53.3.35 CH1D0TSR—Offset E4h ..... 2423
53.3.36 CH1D1SAR—Offset E8h ..... 2423
53.3.37 CH1D1TSR—Offset ECh ..... 2423
53.3.38 CH1D2SAR—Offset F0h ..... 2424
53.3.39 CH1D2TSR—Offset F4h ..... 2424
53.3.40 CH1D3SAR—Offset F8h ..... 2425
53.3.41 CH1D3TSR—Offset FCh ..... 2425
53.4 Registers in I/O Space—IOBA ..... 2426
53.4.1 RBR_THR_DLL—Offset 0h ..... 2426
53.4.2 IER_DLH—Offset 1h ..... 2426
53.4.3 IIR_FCR—Offset 2h ..... 2427
53.4.4 LCR—Offset 3h ..... 2428
53.4.5 MCR—Offset 4h ..... 2430
53.4.6 LSR—Offset 5h ..... 2432
53.4.7 MSR—Offset 6h ..... 2434
53.4.8 SPR—Offset 7h ..... 2436

**54 Innovation Engine - B0, D27, F(0, 1, 3, 4) ..... 2437**
54.1 Introduction and Index ..... 2437
54.1.1 Host Configuration Space—HECI1 (F:0) ..... 2438
54.1.2 Host Configuration Space—HECI2 (F:1) ..... 2440
54.1.3 Host Configuration Space—HECI3 (F:4) ..... 2441
54.1.4 Host Configuration Space—KT (F:3) ..... 2442
54.1.5 Host Configuration Space—IDE (F:2) ..... 2443
54.1.6 Host Memory Space—HECI1_MMIO_MBAR ..... 2444
54.1.7 Host Memory Space—HECI2_MMIO_MBAR ..... 2445
54.1.8 Host Memory Space—HECI3_MMIO_MBAR ..... 2446
54.1.9 Host Memory Space—KT_HOST_MEMBAR ..... 2447
54.1.10 Host Memory Space—Fixed Address ..... 2448
54.1.11 Host I/O Space—KT_HOST_IOBAR ..... 2449
54.1.12 Host I/O Space—IDE_HOST_PCMDIOBAR ..... 2450
54.1.13 Host I/O Space—IDE_HOST_PCTLIOBAR ..... 2451
54.1.14 Host I/O Space—IDE_HOST_BMIOBAR ..... 2452
54.1.15 Sideband Registers ..... 2453
54.2 Registers in Configuration Space—HECI1 ..... 2454
54.2.1 Identifiers (HECI1_ID)—Offset 0h ..... 2454
54.2.2 Command (HECI1_CMD)—Offset 4h ..... 2455
54.2.3 Status (HECI1_STS)—Offset 6h ..... 2456
54.2.4 Revision ID and Class Code (HECI1_RID_CC)—Offset 8h ..... 2457
54.2.5 Cache Line Size (HECI1_CLS)—Offset Ch ..... 2457
54.2.6 Master Latency Timer (HECI1_MLT)—Offset Dh ..... 2457
54.2.7 Header Type (HECI1_HTYPE)—Offset Eh ..... 2458
54.2.8 Built In Self-Test (HECI1_BIST)—Offset Fh ..... 2458
54.2.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h ..... 2458
54.2.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h ..... 2459
54.2.11 Sub System Identifiers (HECI1_SS)—Offset 2Ch ..... 2459
54.2.12 Capabilities Pointer (HECI1_CAP)—Offset 34h ..... 2459
54.2.13 Interrupt Information (HECI1_INTR)—Offset 3Ch ..... 2460
54.2.14 Minimum Grant (HECI1_MGNT)—Offset 3Eh ..... 2460
54.2.15 Maximum Latency (HECI1_MLAT)—Offset 3Fh ..... 2460
54.2.16 Host Firmware Status (HECI1_HFS)—Offset 40h ..... 2461
54.2.17 Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h ..... 2463
54.2.18 General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h ..... 2464

54.2.19 Host General Status (HECI1_H_GS1)—Offset 4Ch ..... 2465
54.2.20 PCI Power Management Capability ID (HECI1_PID)—Offset 50h ..... 2465
54.2.21 PCI Power Management Capabilities (HECI1_PC)—Offset 52h ..... 2465
54.2.22 PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h ..... 2466
54.2.23 General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h ..... 2466
54.2.24 General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h ..... 2467
54.2.25 General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h ..... 2467
54.2.26 General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch ..... 2467
54.2.27 Host General Status 2 (HECI1_H_GS2)—Offset 70h ..... 2467
54.2.28 Host General Status 3 (HECI1_H_GS3)—Offset 74h ..... 2468
54.2.29 Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch ..... 2468
54.2.30 Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh ..... 2468
54.2.31 Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h ..... 2469
54.2.32 Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h ..... 2469
54.2.33 Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h ..... 2469
54.2.34 HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h ..... 2470
54.2.35 Host Extend Register Status (HECI1_HERS)—Offset BCh ..... 2470
54.2.36 Host Extend Register DW1. (HECI1_HER1)—Offset C0h ..... 2471
54.2.37 Host Extend Register DW2. (HECI1_HER2)—Offset C4h ..... 2471
54.2.38 Host Extend Register DW3. (HECI1_HER3)—Offset C8h ..... 2471
54.2.39 Host Extend Register DW4. (HECI1_HER4)—Offset CCh ..... 2471
54.2.40 Host Extend Register DW5. (HECI1_HER5)—Offset D0h ..... 2472
54.2.41 Host Extend Register DW6. (HECI1_HER6)—Offset D4h ..... 2472
54.2.42 Host Extend Register DW7. (HECI1_HER7)—Offset D8h ..... 2472
54.2.43 Host Extend Register DW8. (HECI1_HER8)—Offset DCh ..... 2472
54.2.44 Manufacturer's ID (HECI1_MANID)—Offset F8h ..... 2473
54.3 Registers in Configuration Space—HECI2 ..... 2474
54.3.1 Identifiers (HECI2_ID)—Offset 0h ..... 2474
54.3.2 Command (HECI2_CMD)—Offset 4h ..... 2475
54.3.3 Status (HECI2_STS)—Offset 6h ..... 2476
54.3.4 Revision ID and Class Code (HECI2_RID_CC)—Offset 8h ..... 2477
54.3.5 Cache Line Size (HECI2_CLS)—Offset Ch ..... 2477
54.3.6 Master Latency Timer (HECI2_MLT)—Offset Dh ..... 2477
54.3.7 Header Type (HECI2_HTYPE)—Offset Eh ..... 2478
54.3.8 Built In Self-Test (HECI2_BIST)—Offset Fh ..... 2478
54.3.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h ..... 2478
54.3.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h ..... 2479
54.3.11 Sub System Identifiers (HECI2_SS)—Offset 2Ch ..... 2479
54.3.12 Capabilities Pointer (HECI2_CAP)—Offset 34h ..... 2479
54.3.13 Interrupt Information (HECI2_INTR)—Offset 3Ch ..... 2480
54.3.14 Minimum Grant (HECI2_MGNT)—Offset 3Eh ..... 2480
54.3.15 Maximum Latency (HECI2_MLAT)—Offset 3Fh ..... 2480
54.3.16 Host Firmware Status (HECI2_HFS)—Offset 40h ..... 2481
54.3.17 General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h ..... 2483
54.3.18 Host General Status (HECI2_H_GS1)—Offset 4Ch ..... 2484
54.3.19 PCI Power Management Capability ID (HECI2_PID)—Offset 50h ..... 2484
54.3.20 PCI Power Management Capabilities (HECI2_PC)—Offset 52h ..... 2485
54.3.21 PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h ..... 2486
54.3.22 General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h ..... 2486
54.3.23 General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h ..... 2487
54.3.24 General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h ..... 2487
54.3.25 General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch ..... 2487
54.3.26 Host General Status 2 (HECI2_H_GS2)—Offset 70h ..... 2487

54.3.27 Host General Status 3 (HECI2_H_GS3)—Offset 74h ....2488
54.3.28 Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch ....2488
54.3.29 Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh ....2488
54.3.30 Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h ....2489
54.3.31 Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h ....2489
54.3.32 Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h ....2489
54.3.33 HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h ....2490
54.3.34 Manufacturer's ID (HECI2_MANID)—Offset F8h ....2490
54.4 Registers in Configuration Space—HECI3 ....2491
54.4.1 Identifiers (HECI3_ID)—Offset 0h ....2491
54.4.2 Command (HECI3_CMD)—Offset 4h ....2492
54.4.3 Status (HECI3_STS)—Offset 6h ....2493
54.4.4 Revision ID and Class Code (HECI3_RID_CC)—Offset 8h ....2494
54.4.5 Cache Line Size (HECI3_CLS)—Offset Ch ....2494
54.4.6 Master Latency Timer (HECI3_MLT)—Offset Dh ....2494
54.4.7 Header Type (HECI3_HTYPE)—Offset Eh ....2495
54.4.8 Built In Self-Test (HECI3_BIST)—Offset Fh ....2495
54.4.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h ....2495
54.4.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h ....2496
54.4.11 Sub System Identifiers (HECI3_SS)—Offset 2Ch ....2496
54.4.12 Capabilities Pointer (HECI3_CAP)—Offset 34h ....2496
54.4.13 Interrupt Information (HECI3_INTR)—Offset 3Ch ....2497
54.4.14 Minimum Grant (HECI3_MGNT)—Offset 3Eh ....2497
54.4.15 Maximum Latency (HECI3_MLAT)—Offset 3Fh ....2497
54.4.16 Host Firmware Status (HECI3_HFS)—Offset 40h ....2498
54.4.17 General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h ....2500
54.4.18 Host General Status (HECI3_H_GS1)—Offset 4Ch ....2501
54.4.19 PCI Power Management Capability ID (HECI3_PID)—Offset 50h ....2501
54.4.20 PCI Power Management Capabilities (HECI3_PC)—Offset 52h ....2502
54.4.21 PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h ....2503
54.4.22 General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h ....2503
54.4.23 General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h ....2504
54.4.24 General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h ....2504
54.4.25 General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch ....2504
54.4.26 Host General Status 2 (HECI3_H_GS2)—Offset 70h ....2504
54.4.27 Host General Status 3 (HECI3_H_GS3)—Offset 74h ....2505
54.4.28 Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch ....2505
54.4.29 Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh ....2505
54.4.30 Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h ....2506
54.4.31 Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h ....2506
54.4.32 Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h ....2506
54.4.33 HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h ....2507
54.4.34 Manufacturer's ID (HECI3_MANID)—Offset F8h ....2507
54.5 Registers in Configuration Space—KT (Host) ....2508
54.5.1 Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h ....2508
54.5.2 Status and Command (KT_HOST_STS_CMD)—Offset 4h ....2509
54.5.3 Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h ....2511
54.5.4 BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch ....2512
54.5.5 KT IO BAR (KT_HOST_IOBAR)—Offset 10h ....2512
54.5.6 KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h ....2513
54.5.7 Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h ....2513
54.5.8 Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch ....2514

54.5.9 Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h ..... 2514
54.5.10 Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h ..... 2514
54.5.11 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch ..... 2515
54.5.12 MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h ..... 2516
54.5.13 MSI Message Address (KT_HOST_MSIMA)—Offset 44h ..... 2516
54.5.14 MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h ..... 2517
54.5.15 MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch ..... 2517
54.5.16 Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h ..... 2518
54.5.17 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h ..... 2519
54.5.18 Process Stepping Information (KT_HOST_PSI)—Offset F8h ..... 2520
54.6 Registers in Configuration Space—IDE (Host) ..... 2521
54.6.1 Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h ..... 2521
54.6.2 Status and Command (IDE_HOST_STS_CMD)—Offset 4h..... 2522
54.6.3 Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h ..... 2524
54.6.4 BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch..... 2525
54.6.5 IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h ..... 2525
54.6.6 IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h ..... 2526
54.6.7 IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h ..... 2526
54.6.8 IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch ..... 2526
54.6.9 IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h..... 2527
54.6.10 Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h ..... 2527
54.6.11 Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch ..... 2527
54.6.12 Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h ..... 2528
54.6.13 Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h ..... 2528
54.6.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch..... 2529
54.6.15 MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h..... 2530
54.6.16 MSI Message Address (IDE_HOST_MSIMA)—Offset 44h..... 2530
54.6.17 MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h..... 2531
54.6.18 MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch..... 2531
54.6.19 Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h..... 2532
54.6.20 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h..... 2533
54.6.21 Process Stepping Information (IDE_HOST_PSI)—Offset F8h ..... 2534
54.7 Registers in Memory Space—HECI1_MMIO_MBAR ..... 2535
54.7.1 Host CB Write Window (HECI1_H_CB_WW)—Offset 0h ..... 2535
54.7.2 Host Control and Status Register (HECI1_H_CSR)—Offset 4h ..... 2536
54.7.3 CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h ..... 2537
54.7.4 CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch ..... 2537
54.7.5 D0i3 Control (HECI1_D0I3C)—Offset 800h..... 2538
54.8 Registers in Memory Space—HECI2_MMIO_MBAR ..... 2539

54.8.1 Host CB Write Window (HECI2_H_CB_WW)—Offset 0h ..... 2539
54.8.2 Host Control and Status Register (HECI2_H_CSR)—Offset 4h ..... 2540
54.8.3 CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h ..... 2541
54.8.4 CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch ..... 2541
54.8.5 D0i3 Control (HECI2_D0I3C)—Offset 800h ..... 2542
54.9 Registers in Memory Space—HECI3_MMIO_MBAR ..... 2543
54.9.1 Host CB Write Window (HECI3_H_CB_WW)—Offset 0h ..... 2543
54.9.2 Host Control and Status Register (HECI3_H_CSR)—Offset 4h ..... 2544
54.9.3 CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h ..... 2545
54.9.4 CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch ..... 2545
54.9.5 D0i3 Control (HECI3_D0I3C)—Offset 800h ..... 2546
54.10 Registers in Memory Space—KT_HOST_MEMBAR ..... 2547
54.10.1 KT Receive Buffer Register (MEM_KTRBR)—Offset 0h ..... 2547
54.10.2 KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h ..... 2547
54.10.3 KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h ..... 2547
54.10.4 KT Interrupt Enable register (MEM_KTIER)—Offset 1h ..... 2548
54.10.5 KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h ..... 2548
54.10.6 KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h ..... 2549
54.10.7 KT FIFO Control register (MEM_KTFCR)—Offset 2h ..... 2550
54.10.8 KT Line Control register (MEM_KTLCR)—Offset 3h ..... 2551
54.10.9 KT Modem Control register (MEM_KTMCR)—Offset 4h ..... 2552
54.10.10 KT Line Status register (MEM_KTLSR)—Offset 5h ..... 2553
54.10.11 KT Modem Status register (MEM_KTMSR)—Offset 6h ..... 2554
54.10.12 KT Scratch register (MEM_KTSCR)—Offset 7h ..... 2554
54.11 Registers in Memory Space—Fixed Address ..... 2555
54.11.1 FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h ..... 2555
54.11.2 FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h ..... 2555
54.11.3 FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h ..... 2556
54.11.4 FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch ..... 2556
54.11.5 FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h ..... 2556
54.11.6 FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h ..... 2557
54.11.7 FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h ..... 2557
54.11.8 FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h ..... 2557
54.11.9 Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h ..... 2558
54.11.10 Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h ..... 2558
54.11.11 Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h ..... 2558
54.11.12 Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch ..... 2559
54.11.13 Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h ..... 2559
54.11.14 Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h ..... 2559
54.11.15 Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h ..... 2560
54.11.16 Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch ..... 2560

54.11.17 Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h ..... 2560
54.11.18 Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h ..... 2560
54.11.19 Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h ..... 2561
54.11.20 Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch ..... 2561
54.11.21 FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h ..... 2561
54.11.22 FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h ..... 2561
54.12 Registers in I/O Space—KT_HOST_IOBAR ..... 2562
54.12.1 KT Receive Buffer Register (IO_KTRBR)—Offset 0h ..... 2562
54.12.2 KT Transmitter Holding Register (IO_KTTHR)—Offset 0h ..... 2562
54.12.3 KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h ..... 2563
54.12.4 KT Interrupt Enable register (IO_KTIER)—Offset 1h ..... 2563
54.12.5 KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h ..... 2564
54.12.6 KT Interrupt Identification Register (IO_KTIIR)—Offset 2h ..... 2564
54.12.7 KT FIFO Control register (IO_KTFCR)—Offset 2h ..... 2565
54.12.8 KT Line Control register (IO_KTLCR)—Offset 3h ..... 2566
54.12.9 KT Modem Control register (IO_KTMCR)—Offset 4h ..... 2567
54.12.10 KT Line Status register (IO_KTLSR)—Offset 5h ..... 2568
54.12.11 KT Modem Status register (IO_KTMSR)—Offset 6h ..... 2569
54.12.12 KT Scratch register (IO_KTSCR)—Offset 7h ..... 2569
54.13 Registers in I/O Space—IDE_HOST_PCMDIOBAR ..... 2570
54.13.1 IDE Data Register (IDEDATA)—Offset 0h ..... 2570
54.13.2 IDE Features Register (IDEFR)—Offset 1h ..... 2570
54.13.3 IDE Error Register DEV0 (IDEERD0)—Offset 1h ..... 2571
54.13.4 IDE Error Register DEV1 (IDEERD1)—Offset 1h ..... 2571
54.13.5 IDE Sector Count In Register (IDESCIR)—Offset 2h ..... 2572
54.13.6 IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h ..... 2572
54.13.7 IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h ..... 2572
54.13.8 IDE Sector Number In Register (IDESNIR)—Offset 3h ..... 2573
54.13.9 IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h ..... 2573
54.13.10 IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h ..... 2573
54.13.11 IDE Cylinder Low In Register (IDECLIR)—Offset 4h ..... 2574
54.13.12 IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h ..... 2574
54.13.13 IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h ..... 2575
54.13.14 IDE Cylinder High In Register (IDECHIR)—Offset 5h ..... 2575
54.13.15 IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h ..... 2576
54.13.16 IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h ..... 2576
54.13.17 IDE Drive/Head In Register (IDEDHIR)—Offset 6h ..... 2577
54.13.18 IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h ..... 2577
54.13.19 IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h ..... 2578
54.13.20 IDE Command Register (IDECR)—Offset 7h ..... 2578
54.13.21 IDE Status Register DEV0 (IDESR0)—Offset 7h ..... 2579
54.13.22 IDE Status Register DEV1 (IDESR1)—Offset 7h ..... 2580
54.14 Registers in I/O Space—IDE_HOST_PCTLIOBAR ..... 2581
54.14.1 IDE Device Control Register (IDDCR)—Offset 2h ..... 2581
54.14.2 IDE Alternate Status Register (IDASR)—Offset 2h ..... 2581
54.15 Registers in I/O Space—IDE_HOST_BMIOBAR ..... 2582
54.15.1 IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h ..... 2582

54.15.2 IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h ....2583
54.15.3 IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h ....2584
54.15.4 IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h ....2586
54.15.5 IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h ....2586
54.15.6 IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h ....2586
54.15.7 IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h ....2587
54.15.8 IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h ....2587
54.15.9 IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h ....2588
54.15.10 IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h ....2588
54.15.11 IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah ....2589
54.15.12 IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh ....2590
54.15.13 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch ....2590
54.15.14 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh ....2591
54.15.15 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh ....2591
54.15.16 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh ....2592
54.16 Sideband Registers—KT (Host) ....2593
54.16.1 KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h ....2593
54.16.2 BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h ....2593
54.17 Sideband Registers—IDE (Host) ....2594
54.17.1 IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h ....2594
54.17.2 BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h ....2594

**55 eMMC Controller - B0, D28, F0 ....2595**
55.1 Introduction and Index ....2595
55.1.1 Host Memory Space—BAR or BAR1 ....2596
55.2 Registers in Memory Space—BAR or BAR1 ....2599
55.2.1 SDMA System Address Register/Argument2 Register (SDMASYSADDR)—Offset 0h ....2599
55.2.2 BlockSize Register (BLOCKSIZE)—Offset 4h ....2599
55.2.3 BlockCount Register (BLOCKCOUNT)—Offset 6h ....2599
55.2.4 Argument1 Register (ARGUMENT1)—Offset 8h ....2600
55.2.5 TransferMode Register (TRANSFERMODE)—Offset Ch ....2600
55.2.6 Command Register (COMMAND)—Offset Eh ....2600
55.2.7 Response Register (RESPONSE1)—Offset 10h ....2601
55.2.8 Response Register (RESPONSE2)—Offset 14h ....2602
55.2.9 Response Register (RESPONSE3)—Offset 16h ....2602
55.2.10 Response Register (RESPONSE4)—Offset 18h ....2602
55.2.11 Response Register (RESPONSE5)—Offset 1Ah ....2602
55.2.12 Response Register (RESPONSE6)—Offset 1Ch ....2603
55.2.13 Response Register (RESPONSE7)—Offset 1Eh ....2603
55.2.14 Buffer DataPort Register (DATAPORT)—Offset 20h ....2603
55.2.15 PresentState Register (PRESENTSTATE)—Offset 24h ....2604

55.2.16 HostControl1 Register (HOSTCONTROL1)—Offset 28h ..... 2606
55.2.17 PowerControl Register (POWERCONTROL)—Offset 29h ..... 2607
55.2.18 BlockGapControl Register (BLOCKGAPCONTROL)—Offset 2Ah ..... 2607
55.2.19 Wakeup Control Register (WAKEUPCONTROL)—Offset 2Bh ..... 2608
55.2.20 Clock Control Register (CLOCKCONTROL)—Offset 2Ch ..... 2609
55.2.21 Timeout Control Register (TIMEOUTCONTROL)—Offset 2Eh ..... 2610
55.2.22 Software Reset Register (SOFTWARERESET)—Offset 2Fh ..... 2610
55.2.23 Normal Interrupt Status Register (NORMALINTRSTS)—Offset 30h ..... 2611
55.2.24 ErrorInterruptStatus Register (ERRORINTRSTS)—Offset 32h ..... 2612
55.2.25 Normal Interrupt Status Enable Register (NORMALINTRSTSENA)—Offset 34h ..... 2613
55.2.26 Error Interrupt Status Enable Register (ERRORINTRSTSENA)—Offset 36h ..... 2614
55.2.27 Normal Interrupt Signal Enable Register (NORMALINTRSIGENA)—Offset 38h ..... 2615
55.2.28 Error Interrupt Signal Enable Register (ERRORINTRSIGENA)—Offset 3Ah ..... 2616
55.2.29 Auto CMD12 Error Status Register (AUTOCMDERRSTS)—Offset 3Ch ..... 2617
55.2.30 Host Control2 Register (HOSTCONTROL2)—Offset 3Eh ..... 2618
55.2.31 Capabilities Register (CAPABILITIES)—Offset 40h ..... 2619
55.2.32 Maximum Current Capabilities Register (MAXCURRENTCAP)—Offset 48h .. 2621
55.2.33 Force Event REGISTER for AUTO CMD Error Status (FORCEEVENTFORAUTOCMDERRORSTATUS)—Offset 50h ..... 2621
55.2.34 Force Event Register for Error Interrupt Status (FORCEEVENTFORERRINTSTS)—Offset 52h ..... 2622
55.2.35 ADMA Error Status Register (ADMAERRSTS)—Offset 54h ..... 2623
55.2.36 ADMA System Address Register0&1 (ADMASYSADDR01)—Offset 58h ..... 2623
55.2.37 ADMA System Address Register1 (ADMASYSADDR2)—Offset 5Ch ..... 2623
55.2.38 ADMA System Address Register1 (ADMASYSADDR3)—Offset 5Eh ..... 2624
55.2.39 Preset Value Register for Initialization (PRESETVALUE0)—Offset 60h ..... 2624
55.2.40 Preset Value Register for Default Speed (PRESETVALUE1)—Offset 62h .... 2625
55.2.41 Preset Value Register for High Speed (PRESETVALUE2)—Offset 64h ..... 2625
55.2.42 Preset Value Register for SDR12 (PRESETVALUE3)—Offset 66h ..... 2626
55.2.43 Preset Value Register for SDR25 (PRESETVALUE4)—Offset 68h ..... 2626
55.2.44 Preset Value Register for SDR50 (PRESETVALUE5)—Offset 6Ah ..... 2627
55.2.45 Preset Value Register for SDR104 (PRESETVALUE6)—Offset 6Ch ..... 2627
55.2.46 Preset Value Register for DDR50 (PRESETVALUE7)—Offset 6Eh ..... 2628
55.2.47 Boot Timeout Control Register (BOOTTIMEOUTCNT)—Offset 70h ..... 2628
55.2.48 Preset Value Register for DDR50 (PRESETVALUE8)—Offset 74h ..... 2629
55.2.49 Slot Interrupt Status Register (SLOTINTRSTS)—Offset FCh ..... 2629
55.2.50 Host Controller Version Register (HOSTCONTROLLERVER)—Offset FEh ..... 2629
55.2.51 Software Latency Tolerance Reporting (SW_LTR_VAL)—Offset 804h ..... 2630
55.2.52 Auto Latency Tolerance Reporting (AUTO_LTR_VAL)—Offset 808h ..... 2630
55.2.53 Capabilities Bypass Control Register (CAP_BYPS)—Offset 810h ..... 2631
55.2.54 eMMC Capabilities Bypass Register1 (CAP_BYPS_REG1)—Offset 814h ..... 2632
55.2.55 eMMC Capabilities Bypass Register2 (CAP_BYPS_REG2)—Offset 818h ..... 2634
55.2.56 D0i3 Control Register (REG_D0i3)—Offset 81Ch ..... 2635
55.2.57 Front End Module Tx Command Path Delay Register (Tx_CMD_DLY)—Offset 820h ..... 2635
55.2.58 Front End Module Tx Data Path Delay Register1 (Tx_DATA_DLY_1)—Offset 824h ..... 2636
55.2.59 Front End Module Tx Data Path Delay Register2 (Tx_DATA_DLY_2)—Offset 828h ..... 2637
55.2.60 Front End Module Rx Data Path Delay Register1 (Rx_CMD_DATA_DLY_1)—Offset 82Ch ..... 2638

55.2.61 Front End Module Rx Strobe Path Delay Register (Rx_STROBE_CTRL_PATH)—Offset 830h ....2639
55.2.62 Front End Module Rx Data Path Delay Register2 (Rx_CMD_DATA_DLY_2)—Offset 834h ....2640
55.2.63 Master DLL Software Control Register (MASTER_DLL)—Offset 838h ....2641
55.2.64 Auto Tuning Value (AUTO_TUNING)—Offset 840h....2641
55.2.65 Root Space Select (eMMC_ROOT_SPACE)—Offset 900h....2642

**56 LPC Controller - B0, D31, F0 .... 2643**
56.1 Introduction and Index ....2643
56.1.1 Host Configuration Space....2644
56.2 Registers in Configuration Space ....2645
56.2.1 I/O Decode Ranges (IOD)—Offset 80h....2645
56.2.2 I/O Enables (IOE)—Offset 82h....2646
56.2.3 LPC Generic I/O Range #1 (LGIR1)—Offset 84h....2647
56.2.4 LPC Generic I/O Range #2 (LGIR2)—Offset 88h....2648
56.2.5 LPC Generic I/O Range #3 (LGIR3)—Offset 8Ch....2649
56.2.6 LPC Generic I/O Range #4 (LGIR4)—Offset 90h....2650
56.2.7 USB Legacy Keyboard/Mouse Control (ULKMC)—Offset 94h....2651
56.2.8 LPC Generic Memory Range (LGMR)—Offset 98h....2653
56.2.9 FWH ID Select #1 (FS1)—Offset D0h....2653
56.2.10 FWH ID Select #2 (FS2)—Offset D4h....2654
56.2.11 BIOS Decode Enable (BDE)—Offset D8h....2655
56.2.12 BIOS Control (BC)—Offset DCh....2657
56.2.13 PCI Clock Control (PCCTL)—Offset E0h....2659

**57 Primary to Side Band Bridge - B0, D31, F1 .... 2660**
57.1 Introduction and Index ....2660
57.1.1 Host Configuration Space....2661
57.2 Registers in Configuration Space ....2663
57.2.1 PCI Identifier (PCIID)—Offset 0h....2663
57.2.2 PCI Command (PCICMD)—Offset 4h....2664
57.2.3 PCI Status (PCISTS)—Offset 6h....2665
57.2.4 Revision ID (PCIRID)—Offset 8h....2665
57.2.5 Class Code (PCICC)—Offset 9h....2666
57.2.6 PCI Master Latency Timer (PCIMLT)—Offset Dh....2666
57.2.7 PCI Header Type (PCIHTYPE)—Offset Eh....2666
57.2.8 Sideband Register Access BAR (SBREG_BAR)—Offset 10h....2667
57.2.9 Sideband Register BAR High DWORD (SBREG_BARH)—Offset 14h....2667
57.2.10 PCI Subsystem Identifiers (PCIHSS)—Offset 2Ch....2668
57.2.11 VLW Bus:Device:Function (VBDF)—Offset 50h....2668
57.2.12 ERROR Bus:Device:Function (EBDF)—Offset 52h....2669
57.2.13 Routing Configuration (RCFG)—Offset 54h....2670
57.2.14 High Performance Event Timer Configuration (HPTC)—Offset 60h....2670
57.2.15 IOxAPIC Configuration (IOAC)—Offset 64h....2671
57.2.16 IOxAPIC Bus:Device:Function (IBDF)—Offset 6Ch....2671
57.2.17 HPET Bus:Device:Function (HBDF)—Offset 70h....2672
57.2.18 Sideband Register posted 0 (SBREGPOSTED0)—Offset 80h....2672
57.2.19 Sideband Register posted 1 (SBREGPOSTED1)—Offset 84h....2672
57.2.20 Sideband Register posted 2 (SBREGPOSTED2)—Offset 88h....2673
57.2.21 Sideband Register posted 3 (SBREGPOSTED3)—Offset 8Ch....2673
57.2.22 Sideband Register posted 4 (SBREGPOSTED4)—Offset 90h....2673
57.2.23 Sideband Register posted 5 (SBREGPOSTED5)—Offset 94h....2673
57.2.24 Sideband Register posted 6 (SBREGPOSTED6)—Offset 98h....2674
57.2.25 Sideband Register posted 7 (SBREGPOSTED7)—Offset 9Ch....2674
57.2.26 Display Bus:Device:Function (DISPBDF)—Offset A0h....2674

57.2.27 ICC Register Offsets (ICCOS)—Offset A4h ..... 2675
57.2.28 Endpoint Mask 0 (EPMASK0)—Offset B0h ..... 2675
57.2.29 Endpoint Mask 1 (EPMASK1)—Offset B4h ..... 2675
57.2.30 Endpoint Mask 2 (EPMASK2)—Offset B8h ..... 2676
57.2.31 Endpoint Mask 3 (EPMASK3)—Offset BCh ..... 2676
57.2.32 Endpoint Mask 4 (EPMASK4)—Offset C0h ..... 2676
57.2.33 Endpoint Mask 5 (EPMASK5)—Offset C4h ..... 2676
57.2.34 Endpoint Mask 6 (EPMASK6)—Offset C8h ..... 2677
57.2.35 Endpoint Mask 7 (EPMASK7)—Offset CCh ..... 2677
57.2.36 SBI Address (SBIADDR)—Offset D0h ..... 2677
57.2.37 SBI Data (SBIDATA)—Offset D4h ..... 2678
57.2.38 SBI Status (SBISTAT)—Offset D8h ..... 2678
57.2.39 SBI Routing Identification (SBIRID)—Offset DAh ..... 2679
57.2.40 SBI Extended Address (SBIEXTADDR)—Offset DCh ..... 2679
57.2.41 P2SB Control (P2SBC)—Offset E0h ..... 2680
57.2.42 Unsupported Request Error Status (URES)—Offset F0h ..... 2681
57.2.43 Unsupported Request Error Control (UREC)—Offset F4h ..... 2681
57.2.44 Manufacturer's ID (MANID)—Offset F8h ..... 2682

**58 Power Management Controller - B0, D31, F2 ..... 2683**
58.1 Introduction and Index ..... 2683
58.1.1 Host Configuration Space ..... 2684
58.1.2 Host Memory Space—PWRMBASE ..... 2685
58.1.3 Host I/O Space—ABASE ..... 2686
58.1.4 Host I/O Space—Fixed Address ..... 2687
58.1.5 Sideband Registers ..... 2688
58.2 Registers in Configuration Space ..... 2689
58.2.1 PCI Identifier (PCIID)—Offset 0h ..... 2689
58.2.2 ACPI Base Address (ABASE)—Offset 40h ..... 2689
58.2.3 ACPI Control (ACTL)—Offset 44h ..... 2690
58.2.4 General PM Configuration A (GEN_PMCON_A)—Offset A0h ..... 2691
58.2.5 General PM Configuration B (GEN_PMCON_B)—Offset A4h ..... 2693
58.2.6 Extended Test Mode Register 3 (ETR3)—Offset ACh ..... 2695
58.3 Registers in Memory Space—PWRMBASE ..... 2696
58.3.1 Wake Alarm Device Timer: AC (WADT_AC)—Offset 0h ..... 2696
58.3.2 Wake Alarm Device Timer: DC (WADT_DC)—Offset 4h ..... 2697
58.3.3 Wake Alarm Device Expired Timer: AC (WADT_EXP_AC)—Offset 8h ..... 2698
58.3.4 Wake Alarm Device Expired Timer: DC (WADT_EXP_DC)—Offset Ch ..... 2698
58.3.5 Power and Reset Status (PRSTS)—Offset 10h ..... 2699
58.3.6 Power Management Configuration Reg 1 (PM_CFG)—Offset 18h ..... 2700
58.3.7 PCH Power Management Status (PCH_PM_STS)—Offset 1Ch ..... 2704
58.3.8 PCH Power Management Status (PCH_PM_STS2)—Offset 24h ..... 2705
58.3.9 ADR Enable (ADR_EN)—Offset F0h ..... 2707
58.3.10 System Time Capture Trigger (SYSTIMCT)—Offset F8h ..... 2709
58.3.11 ACPI Timer Control (ACPI_TMR_CTL)—Offset FCh ..... 2709
58.3.12 GPIO Configuration (GPIO_CFG)—Offset 120h ..... 2710
58.3.13 Global Reset Causes (GBLRST_CAUSE0)—Offset 124h ..... 2711
58.3.14 Global Reset Causes (GBLRST_CAUSE1)—Offset 128h ..... 2713
58.3.15 Host Partition Reset Causes (HPR_CAUSE0)—Offset 12Ch ..... 2714
58.3.16 modPHY Power Management Configuration Reg 1 (MODPHY_PM_CFG1)—Offset 200h ..... 2715
58.3.17 modPHY Power Management Configuration Reg 2 (MODPHY_PM_CFG2)—Offset 204h ..... 2716
58.3.18 modPHY Power Management Configuration Reg 3 (MODPHY_PM_CFG3)—Offset 208h ..... 2718

58.3.19 modPHY Power Management Configuration Reg 4 (MODPHY_PM_CFG4)—Offset 20Ch ....2719
58.3.20 IE_STS (IE_STS)—Offset 3F0h ....2720
58.3.21 ME_STS (ME_STS)—Offset 3F4h ....2722
58.3.22 Static PG Related Function Disable Register 1 (ST_PG_FDIS_PMC_1)—Offset 620h ....2723
58.3.23 Non-Static PG Related Function Disable Register 1 (NST_PG_FDIS_1)—Offset 628h ....2724
58.3.24 Non-Static PG Fuse Disable Read 1 Register (N_STPG_FUSE_SS_DIS_RD_1)—Offset 640h ....2724
58.3.25 Static PG Fuse and Soft Strap Disable Read Register 2 (STPG_FUSE_SS_DIS_RD_2)—Offset 644h ....2725
58.4 Registers in I/O Space—ABASE ....2726
58.4.1 Power Management 1 Enables and Status (PM1_EN_STS)—Offset 0h ....2726
58.4.2 Power Management 1 Control —Offset 4h ....2728
58.4.3 Power Management 1 Timer (PM1_TMR)—Offset 8h ....2729
58.4.4 SMI Control and Enable (SMI_EN)—Offset 30h ....2730
58.4.5 SMI Status Register (SMI_STS)—Offset 34h ....2732
58.4.6 General Purpose Event Control (GPE_CTRL)—Offset 40h ....2735
58.4.7 Device Trap Status Register (DEVTRAP_STS)—Offset 44h ....2736
58.4.8 PM2a Control Block (PM2A_CNT_BLK)—Offset 50h ....2737
58.4.9 Over-Clocking WDT Control (OC_WDT_CTL)—Offset 54h ....2737
58.4.10 General Purpose Event 0 Status [31:0] (GPE0_STS_31_0)—Offset 80h ....2738
58.4.11 General Purpose Event 0 Status [63:32] (GPE0_STS_63_32)—Offset 84h ....2739
58.4.12 General Purpose Event 0 Status [95:64] (GPE0_STS_95_64)—Offset 88h ....2739
58.4.13 General Purpose Event 0 Status [127:96] (GPE0_STS_127_96)—Offset 8Ch ....2740
58.4.14 General Purpose Event 0 Enable [31:0] (GPE0_EN_31_0)—Offset 90h ....2742
58.4.15 General Purpose Event 0 Enable [63:32] (GPE0_EN_63_32)—Offset 94h ....2742
58.4.16 General Purpose Event 0 Enable [95:64] (GPE0_EN_95_64)—Offset 98h ....2742
58.4.17 General Purpose Event 0 Enable [127:96] (GPE0_EN_127_96)—Offset 9Ch ....2743
58.5 Registers in I/O Space—Fixed Addresses ....2745
58.5.1 Reset Control Register (RST_CNT) ....2745
58.6 Sideband Registers ....2746
58.6.1 Advanced Power Management Status and Control Port (APM_STS_CNT)—Offset B0h ....2746
58.6.2 Misc SB Reg0 (MISC_SBR0)—Offset B4h ....2746

**59 SMBus - Legacy - B0, D31, F4 ....2748**
59.1 Introduction and Index ....2748
59.1.1 Registers in Configuration Space ....2749
59.1.2 Registers in Memory Space—SMBMBAR ....2750
59.1.3 Registers in I/O Space—SBA ....2751
59.1.4 Registers in I/O Space—TCOBASE ....2752
59.1.5 Sideband Registers—Host SMBus Private Configuration ....2753
59.1.6 Sideband Registers—CLTT SMBus Configuration Control ....2754
59.2 Registers in Configuration Space ....2755
59.2.1 Vendor ID (VID)—Offset 0h ....2755
59.2.2 Device ID (DID)—Offset 2h ....2755
59.2.3 Command (CMD)—Offset 4h ....2756

59.2.4 Device Status (DS)—Offset 6h ..... 2757
59.2.5 Revision ID (RID)—Offset 8h ..... 2758
59.2.6 Programming Interface (PI)—Offset 9h ..... 2758
59.2.7 Sub Class Code (SCC)—Offset Ah ..... 2758
59.2.8 Base Class Code (BCC)—Offset Bh ..... 2758
59.2.9 SMBus Memory Base Address_31_0 (SMBMBAR_31_0)—Offset 10h ..... 2759
59.2.10 SMBus Memory Base Address_63_32 (SMBMBAR_63_32)—Offset 14h ..... 2759
59.2.11 SMB Base Address (SBA)—Offset 20h ..... 2760
59.2.12 SVID—Offset 2Ch ..... 2760
59.2.13 SID—Offset 2Eh ..... 2760
59.2.14 Interrupt Line (INTLN)—Offset 3Ch ..... 2761
59.2.15 Interrupt Pin (INTPN)—Offset 3Dh ..... 2761
59.2.16 Host Configuration (HCFG)—Offset 40h ..... 2762
59.2.17 TCO Base Address (TCOBASE)—Offset 50h ..... 2763
59.2.18 TCO Control (TCOCTL)—Offset 54h ..... 2763
59.2.19 Host_Timing (HTIM)—Offset 64h ..... 2764
59.2.20 Manufacturer's ID (MANID)—Offset F8h ..... 2765
59.3 Registers in Memory Space—SMBMBAR ..... 2766
59.3.1 Host Status Register Address (HSTS)—Offset 0h ..... 2766
59.3.2 Host Control Register (HCTL)—Offset 2h ..... 2767
59.3.3 Host Command Register (HCMD)—Offset 3h ..... 2769
59.3.4 Transmit Slave Address Register (TSA)—Offset 4h ..... 2769
59.3.5 Data 0 Register (HD0)—Offset 5h ..... 2769
59.3.6 Data 1 Register (HD1)—Offset 6h ..... 2770
59.3.7 Host Block Data (HBD)—Offset 7h ..... 2770
59.3.8 Packet Error Check Data Register (PEC)—Offset 8h ..... 2771
59.3.9 Receive Slave Address Register (RSA)—Offset 9h ..... 2771
59.3.10 Slave Data Register (SD)—Offset Ah ..... 2772
59.3.11 Auxiliary Status (AUXS)—Offset Ch ..... 2773
59.3.12 Auxiliary Control (AUXC)—Offset Dh ..... 2774
59.3.13 SMLINK_PIN_CTL Register (SMLC)—Offset Eh ..... 2775
59.3.14 SMBUS_PIN_CTL Register (SMBC)—Offset Fh ..... 2776
59.3.15 Slave Status Register (SSTS)—Offset 10h ..... 2777
59.3.16 Slave Command Register (SCMD)—Offset 11h ..... 2778
59.3.17 Notify Device Address Register (NDA)—Offset 14h ..... 2779
59.3.18 Notify Data Low Byte Register (NDLB)—Offset 16h ..... 2779
59.3.19 Notify Data High Byte Register (NDHB)—Offset 17h ..... 2779
59.4 Registers in I/O Space—SBA ..... 2780
59.4.1 Host Status Register Address (HSTS)—Offset 0h ..... 2780
59.4.2 Host Control Register (HCTL)—Offset 2h ..... 2781
59.4.3 Host Command Register (HCMD)—Offset 3h ..... 2783
59.4.4 Transmit Slave Address Register (TSA)—Offset 4h ..... 2783
59.4.5 Data 0 Register (HD0)—Offset 5h ..... 2783
59.4.6 Data 1 Register (HD1)—Offset 6h ..... 2784
59.4.7 Host Block Data (HBD)—Offset 7h ..... 2784
59.4.8 Packet Error Check Data Register (PEC)—Offset 8h ..... 2785
59.4.9 Receive Slave Address Register (RSA)—Offset 9h ..... 2785
59.4.10 Slave Data Register (SD)—Offset Ah ..... 2786
59.4.11 Auxiliary Status (AUXS)—Offset Ch ..... 2787
59.4.12 Auxiliary Control (AUXC)—Offset Dh ..... 2788
59.4.13 SMLINK_PIN_CTL Register (SMLC)—Offset Eh ..... 2789
59.4.14 SMBUS_PIN_CTL Register (SMBC)—Offset Fh ..... 2790
59.4.15 Slave Status Register (SSTS)—Offset 10h ..... 2791
59.4.16 Slave Command Register (SCMD)—Offset 11h ..... 2792
59.4.17 Notify Device Address Register (NDA)—Offset 14h ..... 2793

59.4.18 Notify Data Low Byte Register (NDLB)—Offset 16h ....2793
59.4.19 Notify Data High Byte Register (NDHB)—Offset 17h ....2793
59.5 Registers in I/O Space—TCOBASE ....2794
59.5.1 TCO_RLD Register (TRLD)—Offset 0h ....2794
59.5.2 TCO_DAT_IN Register (TDI)—Offset 2h ....2794
59.5.3 TCO_DAT_OUT Register (TDO)—Offset 3h ....2795
59.5.4 TCO1_STS Register (TSTS1)—Offset 4h ....2795
59.5.5 TCO2_STS Register (TSTS2)—Offset 6h ....2797
59.5.6 TCO1_CNT Register (TCTL1)—Offset 8h ....2798
59.5.7 TCO2_CNT Register (TCTL2)—Offset Ah ....2799
59.5.8 LEGACY_ELIM Register (LE)—Offset 10h ....2800
59.5.9 TCO_TMR Register (TTMR)—Offset 12h ....2800
59.6 Sideband Registers—Host SMBus Private Configuration ....2801
59.6.1 TCO Configuration (TCOCFG)—Offset 0h ....2801
59.6.2 General Control (GC)—Offset Ch ....2802
59.6.3 Power Control Enable (PCE)—Offset 10h ....2803
59.7 Sideband Registers—CLTT SMBus Configuration Control ....2804
59.7.1 CLTT Status (CLTT_STS)—Offset 0h ....2804
59.7.2 Ch0 Dimm0 Cfg Status (C0D0CFGSTS)—Offset 10h ....2805
59.7.3 Ch0 Dimm1 Cfg Status (C0D1CFGSTS)—Offset 14h ....2805
59.7.4 Ch0 Dimm2 Cfg Status (C0D2CFGSTS)—Offset 18h ....2806
59.7.5 Ch0 Dimm3 Cfg Status (C0D3CFGSTS)—Offset 1Ch ....2806
59.7.6 Ch1 Dimm0 Cfg Status (C1D0CFGSTS)—Offset 20h ....2807
59.7.7 Ch1 Dimm1 Cfg Status (C1D1CFGSTS)—Offset 24h ....2807
59.7.8 Ch1 Dimm2 Cfg Status (C1D2CFGSTS)—Offset 28h ....2808
59.7.9 Ch1 Dimm3 Cfg Status (C1D3CFGSTS)—Offset 2Ch ....2808
59.7.10 Ch0 Dimm Temp Status (C0TEMPSTS0)—Offset 30h ....2809
59.7.11 Ch0 Dimm Temp Status (C0TEMPSTS1)—Offset 34h ....2809
59.7.12 Ch1 Dimm Temp Status (C1TEMPSTS0)—Offset 38h ....2809
59.7.13 Ch1 Dimm Temp Status (C1TEMPSTS1)—Offset 3Ch ....2810
59.7.14 Host Configuration (HCFG)—Offset 40h ....2810

**60 SPI Controller - B0, D31, F5 ....2811**
60.1 Introduction and Index ....2811
60.1.1 Host Configuration Space ....2812
60.1.2 Host Memory Space—BIOS_SPI_BAR0 ....2813
60.2 Registers in Configuration Space ....2815
60.2.1 Device ID and Vendor ID (BIOS_SPI_DID_VID)—Offset 0h ....2815
60.2.2 Status and Command (BIOS_SPI_STS_CMD)—Offset 4h ....2816
60.2.3 Revision ID (BIOS_SPI_CC_RID)—Offset 8h ....2818
60.2.4 BIST, Header Type, Latency Timer, Cache Line Size (BIOS_SPI_BIST_HTYPE_LT_CLS)—Offset Ch ....2818
60.2.5 SPI BAR0 MMIO (BIOS_SPI_BAR0)—Offset 10h ....2819
60.2.6 Cardbus CIS Pointer (BIOS_SPI_CCP)—Offset 28h ....2819
60.2.7 Subsystem and Vendor ID (BIOS_SPI_SID_SVID)—Offset 2Ch ....2819
60.2.8 Expansion ROM Base Address (BIOS_SPI_XRBAR)—Offset 30h ....2820
60.2.9 Capabilities List Pointer (BIOS_SPI_CAPP)—Offset 34h ....2820
60.2.10 SPI Unsupported Request Status (BIOS_SPI_UR_STS_CTL)—Offset D0h ....2820
60.2.11 BIOS Decode Enable (BIOS_SPI_BDE)—Offset D8h ....2821
60.2.12 BIOS Control (BIOS_SPI_BC)—Offset DCh ....2823
60.3 Registers in Memory Space—BIOS_SPI_BAR0 ....2825
60.3.1 SPI BIOS MMIO PRI (BIOS_BFPREG)—Offset 0h ....2825
60.3.2 Hardware Sequencing Flash Status and Control (BIOS_HSFSTS_CTL)—Offset 4h ....2826
60.3.3 Flash Address (BIOS_FADDR)—Offset 8h ....2828

60.3.4 Discrete Lock Bits (BIOS_DLOCK)—Offset Ch ..... 2829
60.3.5 Flash Data 0 (BIOS_FDATA0)—Offset 10h..... 2830
60.3.6 Flash Data 1 (BIOS_FDATA1)—Offset 14h..... 2830
60.3.7 Flash Data 2 (BIOS_FDATA2)—Offset 18h..... 2830
60.3.8 Flash Data 3 (BIOS_FDATA3)—Offset 1Ch ..... 2831
60.3.9 Flash Data 4 (BIOS_FDATA4)—Offset 20h..... 2831
60.3.10 Flash Data 5 (BIOS_FDATA5)—Offset 24h..... 2831
60.3.11 Flash Data 6 (BIOS_FDATA6)—Offset 28h..... 2832
60.3.12 Flash Data 7 (BIOS_FDATA7)—Offset 2Ch ..... 2832
60.3.13 Flash Data 8 (BIOS_FDATA8)—Offset 30h..... 2832
60.3.14 Flash Data 9 (BIOS_FDATA9)—Offset 34h..... 2833
60.3.15 Flash Data 10 (BIOS_FDATA10)—Offset 38h ..... 2833
60.3.16 Flash Data 11 (BIOS_FDATA11)—Offset 3Ch ..... 2833
60.3.17 Flash Data 12 (BIOS_FDATA12)—Offset 40h ..... 2834
60.3.18 Flash Data 13 (BIOS_FDATA13)—Offset 44h ..... 2834
60.3.19 Flash Data 14 (BIOS_FDATA14)—Offset 48h ..... 2834
60.3.20 Flash Data 15 (BIOS_FDATA15)—Offset 4Ch ..... 2834
60.3.21 Flash Region Access Permissions (BIOS_FRACC)—Offset 50h..... 2835
60.3.22 Flash Region 0 (BIOS_FREG0)—Offset 54h ..... 2836
60.3.23 Flash Region 1 (BIOS_FREG1)—Offset 58h ..... 2836
60.3.24 Flash Region 2 (BIOS_FREG2)—Offset 5Ch ..... 2837
60.3.25 Flash Region 3 (BIOS_FREG3)—Offset 60h ..... 2837
60.3.26 Flash Region 4 (BIOS_FREG4)—Offset 64h ..... 2838
60.3.27 Flash Region 5 (BIOS_FREG5)—Offset 68h ..... 2838
60.3.28 Flash Region 6 (BIOS_FREG6)—Offset 6Ch ..... 2839
60.3.29 Flash Region 7 (BIOS_FREG7)—Offset 70h ..... 2839
60.3.30 Flash Region 8 (BIOS_FREG8)—Offset 74h ..... 2840
60.3.31 Flash Region 9 (BIOS_FREG9)—Offset 78h ..... 2840
60.3.32 Flash Region 10 (BIOS_FREG10)—Offset 7Ch..... 2841
60.3.33 Flash Region 11 (BIOS_FREG11)—Offset 80h ..... 2841
60.3.34 Flash Protected Range 0 (BIOS_FPR0)—Offset 84h ..... 2842
60.3.35 Flash Protected Range 1 (BIOS_FPR1)—Offset 88h ..... 2843
60.3.36 Flash Protected Range 2 (BIOS_FPR2)—Offset 8Ch ..... 2844
60.3.37 Flash Protected Range 3 (BIOS_FPR3)—Offset 90h ..... 2845
60.3.38 Flash Protected Range 4 (BIOS_FPR4)—Offset 94h ..... 2846
60.3.39 Global Protected Range 0 (BIOS_GPR0)—Offset 98h ..... 2847
60.3.40 Software Sequencing Flash Status and Control (BIOS_SSFSTS_CTL)—Offset A0h ..... 2848
60.3.41 Prefix Opcode and Opcode Type Configuration (BIOS_PREOP_OPTYPE)—Offset A4h ..... 2850
60.3.42 Opcode Menu0 Configuration (BIOS_OPMENU0)—Offset A8h ..... 2851
60.3.43 Opcode Menu1 Configuration (BIOS_OPMENU1)—Offset ACh ..... 2852
60.3.44 Secondary Flash Region Access Permissions (BIOS_SFRACC)—Offset B0h ..... 2852
60.3.45 Flash Descriptor Observability Control (BIOS_FDOC)—Offset B4h..... 2853
60.3.46 Flash Descriptor Observability Data (BIOS_FDOD)—Offset B8h ..... 2854
60.3.47 Additional Flash Control (BIOS_AFC)—Offset C0h..... 2854
60.3.48 Vendor Specific Component Capabilities for Component 0 (BIOS_SFDP0_VSCC0)—Offset C4h..... 2855
60.3.49 Vendor Specific Component Capabilities for Component 1 (BIOS_SFDP1_VSCC1)—Offset C8h..... 2858
60.3.50 Parameter Table Index (BIOS_PTINX)—Offset CCh ..... 2860
60.3.51 Parameter Table Data (BIOS_PTDATA)—Offset D0h ..... 2861
60.3.52 SPI Bus Requester Status (BIOS_SBRS)—Offset D4h ..... 2862
60.3.53 Flash Region 12 (BIOS_FREG12)—Offset E0h ..... 2863

60.3.54 Flash Region 13 (BIOS_FREG13)—Offset E4h....2863
60.3.55 Flash Region 14 (BIOS_FREG14)—Offset E8h....2864
60.3.56 Flash Region 15 (BIOS_FREG15)—Offset ECh....2864
60.3.57 Set Strap Msg Lock (SSML)—Offset F0h....2865
60.3.58 Set Strap Msg Control (SSMC)—Offset F4h....2865
60.3.59 Set Strap Msg Data (SSMD)—Offset F8h....2866
60.3.60 RPMC SFDP Table (BIOS_RPMC0_D0)—Offset 108h....2866
60.3.61 RPMC SFDP Table (BIOS_RPMC1_D0)—Offset 10Ch....2867
60.3.62 RPMC SFDP Table (BIOS_RPMC0_D1)—Offset 110h....2867
60.3.63 RPMC SFDP Table (BIOS_RPMC1_D1)—Offset 114h....2867
60.3.64 BIOS Master Read Access Permissions (BIOS_BM_RAP)—Offset 118h....2868
60.3.65 BIOS Master Write Access Permissions (BIOS_BM_WAP)—Offset 11Ch....2868
60.3.66 IE Flash Protected Range 0 (BIOS_IE_PR0)—Offset 1B0h....2869
60.3.67 IE Flash Protected Range 1 (BIOS_IE_PR1)—Offset 1B4h....2870
60.3.68 IE Flash Protected Range 2 (BIOS_IE_PR2)—Offset 1B8h....2871
60.3.69 IE Flash Protected Range 3 (BIOS_IE_PR3)—Offset 1BCh....2872
60.3.70 IE Flash Protected Range 4 (BIOS_IE_PR4)—Offset 1C0h....2873

**61 Programmable Interval Timer - Legacy....2874**
61.1 Introduction and Index....2874
61.1.1 Host I/O Space—Fixed Address....2874
61.2 Registers in I/O Space—Fixed Addresses....2875
61.2.1 Counter 0 - Interval Timer Status Byte Format Register (C0_ITSBFR)—Offset 40h....2876
61.2.2 Counter 2 - Interval Timer Status Byte Format Register (C2_ITSBFR)—Offset 42h....2877
61.2.3 Timer Control Word Register (TCW)—Offset 43h....2878
61.2.4 Counter 2 - Counter Access Ports Register (C2_CAPR)—Offset 42h....2879
61.2.5 Read Back Command (RBC)—Offset 43h....2880
61.2.6 Counter Latch Command (CLC)—Offset 43h....2881
61.2.7 NMI Status and Control (NMI_STS_CNT)—Offset 61h....2882

**62 Real Time Clock - Legacy....2883**
62.1 Introduction and Index....2883
62.1.1 Host I/O Space—Fixed Address....2884
62.1.2 Host I/O Space—RTC Standard Static RAM Bank (Indexed Data)....2885
62.1.3 Host I/O Space—RTC Extended Static RAM Bank (Indexed Data)....2886
62.1.4 Sideband Registers....2887
62.2 Registers in I/O Space—Fixed Addresses....2888
62.2.1 NMI Enable (and Real Time Clock Index) (NMI_EN)—Port 70h....2888
62.2.2 Real Time Clock Data—Port 71h....2889
62.2.3 Real Time Clock Index to Extended RAM—Port 72h....2889
62.2.4 Real Time Clock Data—Port 73h....2889
62.3 Registers in Host I/O Space—RTC Standard Static RAM Bank (Indexed Data)....2890
62.3.1 Seconds (Sec)—Index 0h....2890
62.3.2 Seconds Alarm (Sec_Alarm)—Index 1h....2890
62.3.3 Minutes—Index 2h....2890
62.3.4 Minutes Alarm (Minutes_Alarm)—Index 3h....2891
62.3.5 Hours—Index 4h....2891
62.3.6 Hours Alarm (Hours_Alarm)—Index 5h....2891
62.3.7 Day of Week (Day_of_Week)—Index 6h....2892
62.3.8 Day of Month (Day_of_Month)—Index 7h....2892
62.3.9 Month—Index 8h....2892
62.3.10 Year—Index 9h....2893
62.3.11 Register A (Register_A)—Index Ah....2894
62.3.12 Register B - General Configuration (Register_B)—Index Bh....2895

62.3.13 Register C - Flag Register (Register_C)—Index Ch ..... 2897
62.3.14 Register D - Flag Register (Register_D)—Index Dh ..... 2898
62.3.15 User RAM—Index Eh through 7Fh ..... 2898
62.4 Registers in Host I/O Space—RTC Extended Static RAM Bank (Indexed Data) ..... 2899
62.4.1 User RAM—Index 0h through 7Fh ..... 2899
62.5 Sideband Registers ..... 2900
62.5.1 RTC Configuration (RC)—Offset 3400h ..... 2900
62.5.2 Backed Up Control (BUC)—Offset 3414h ..... 2901
62.5.3 RTC Update In Progress SMI Control (UIPSMI)—Offset 3F04h ..... 2902
62.5.4 RTC Dynamic Clock Gating Control (RTCDCG)—Offset 3418h ..... 2903

**63 High Precision Event Timer - Legacy ..... 2904**
63.1 Introduction and Index ..... 2904
63.1.1 Host Memory Space—Fixed Address ..... 2905
63.2 Registers in Memory Space—Fixed Addresses ..... 2906
63.2.1 General Config Register (GEN_CFG)—Offset 10h ..... 2906
63.2.2 General Interrupt Status Register (GEN_INT_STS)—Offset 20h ..... 2907

**64 PIC and I/O APIC Controllers ..... 2908**
64.1 Introduction and Index ..... 2908
64.1.1 Host Configuration Space ..... 2909
64.1.2 Host Memory Space—Fixed Address ..... 2910
64.1.3 Host I/O Space—Fixed Address ..... 2912
64.2 Registers in Configuration Space ..... 2913
64.2.1 IOxAPIC Configuration (IOAC)—Offset 64h ..... 2913
64.3 Registers in Memory Space—Fixed Addresses ..... 2914
64.3.1 Index Register (IDX)—FEC0_0000h ..... 2914
64.3.2 Window Register (WDW)—FEC0_0010h ..... 2914
64.3.3 IOxAPIC Interrupt Messages for Local APIC—FEE0_0000h ..... 2914
64.4 Registers Accessed through IOxAPIC IDX and WDW ..... 2916
64.4.1 Version (VS)—Offset 1h ..... 2917
64.4.2 Redirection Table Entry 0 (RTE0)—Offset 10h ..... 2918
64.4.3 Redirection Table Entry 1 (RTE1)—Offset 12h ..... 2920
64.4.4 Redirection Table Entry 2 (RTE2)—Offset 14h ..... 2922
64.4.5 Redirection Table Entry 118 (RTE118)—Offset FCh ..... 2924
64.4.6 Redirection Table Entry 119 (RTE119)—Offset FEh ..... 2926
64.5 Registers in I/O Space—Fixed Addresses ..... 2928
64.5.1 Master Initialization Command Word 1 (MICW1)—I/O Port 20h ..... 2928
64.5.2 Master Initialization Command Word 2 (MICW2)—I/O Port 21h ..... 2929
64.5.3 Master Initialization Command Word 3 (MICW3)—I/O Port 21h ..... 2929
64.5.4 Master Initialization Command Word 4 (MICW4)—I/O Port 21h ..... 2930
64.5.5 Master Operational Control Word 1 (MOCW1)—I/O Port 21h ..... 2930
64.5.6 Master Operational Control Word 2 (MOCW2)—I/O Port 20h ..... 2931
64.5.7 Master Operational Control Word 3 (MOCW3)—I/O Port 20h ..... 2932
64.5.8 Slave Initialization Command Word 1 (SICW1)—I/O Port A0h ..... 2933
64.5.9 Slave Initialization Command Word 2 (SICW2)—I/O Port A1h ..... 2934
64.5.10 Slave Initialization Command Word 3 (SICW3)—I/O Port A1h ..... 2934
64.5.11 Slave Initialization Command Word 4 (SICW4)—I/O Port A1h ..... 2935
64.5.12 Slave Operational Control Word 1 (SOCW1)—I/O Port A1h ..... 2935
64.5.13 Slave Operational Control Word 2 (SOCW2)—I/O Port A0h ..... 2936
64.5.14 Slave Operational Control Word 3 (SOCW3)—I/O Port A0h ..... 2937
64.5.15 Master Edge/Level Control (ELCR1)—I/O Port 4D0h ..... 2938
64.5.16 Slave Edge/Level Control (ELCR2)—I/O Port 4D1h ..... 2938

**65 Customer General-Purpose I/O Controller ..... 2939**

65.1 Introduction and Index ....2939
65.1.1 Sideband Registers—Configurable I/O Signals ....2940
65.2 Sideband Registers, Part 1 of 2—Sideband Port 0xC2 ....2953
65.2.1 Pad Base Address (PADBAR)—Offset Ch ....2953
65.2.2 Miscellaneous Configuration (MISCCFG)—Offset 10h ....2954
65.2.3 Pad Ownership (PAD_OWN_NORTH_ALL_0)—Offset 20h ....2955
65.2.4 Pad Ownership (PAD_OWN_NORTH_ALL_1)—Offset 24h ....2957
65.2.5 Pad Ownership (PAD_OWN_NORTH_ALL_2)—Offset 28h ....2959
65.2.6 Pad Ownership (PAD_OWN_NORTH_ALL_3)—Offset 2Ch ....2961
65.2.7 Pad Ownership (PAD_OWN_NORTH_ALL_4)—Offset 30h ....2963
65.2.8 Pad Ownership (PAD_OWN_NORTH_ALL_5)—Offset 34h ....2965
65.2.9 Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_0)—Offset 90h ....2966
65.2.10 Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_0)—Offset 94h ....2968
65.2.11 Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_1)—Offset 98h ....2970
65.2.12 Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_1)—Offset 9Ch ....2971
65.2.13 Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_0)—Offset C0h ....2972
65.2.14 Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_1)—Offset C4h ....2974
65.2.15 GPI Interrupt Status (GPI_IS_NORTH_ALL_0)—Offset 100h ....2975
65.2.16 GPI Interrupt Status (GPI_IS_NORTH_ALL_1)—Offset 104h ....2977
65.2.17 GPI Interrupt Enable (GPI_IE_NORTH_ALL_0)—Offset 120h ....2978
65.2.18 GPI Interrupt Enable (GPI_IE_NORTH_ALL_1)—Offset 124h ....2980
65.2.19 Reserved[0-5] (RSVD6[0-5])—Offset 128h, Count 6, Stride 4h ....2981
65.2.20 GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_0)—Offset 140h ....2981
65.2.21 GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_1)—Offset 144h ....2983
65.2.22 GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_0)—Offset 160h ....2984
65.2.23 GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_1)—Offset 164h ....2986
65.2.24 SMI Status (GPI_SMI_STS_NORTH_ALL_0)—Offset 180h ....2987
65.2.25 SMI Status (GPI_SMI_STS_NORTH_ALL_1)—Offset 184h ....2989
65.2.26 SMI Enable (GPI_SMI_EN_NORTH_ALL_0)—Offset 1A0h ....2990
65.2.27 SMI Enable (GPI_SMI_EN_NORTH_ALL_1)—Offset 1A4h ....2992
65.2.28 NMI Status (GPI_NMI_STS_NORTH_ALL_0)—Offset 1C0h ....2993
65.2.29 NMI Status (GPI_NMI_STS_NORTH_ALL_1)—Offset 1C4h ....2995
65.2.30 NMI Enable (GPI_NMI_EN_NORTH_ALL_0)—Offset 1E0h ....2996
65.2.31 NMI Enable (GPI_NMI_EN_NORTH_ALL_1)—Offset 1E4h ....2998
65.2.32 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP0)—Offset 400h ....2999
65.2.33 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP0)—Offset 404h ....3002
65.2.34 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP0)—Offset 408h ....3003
65.2.35 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP0)—Offset 40Ch ....3006
65.2.36 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP1)—Offset 410h ....3007
65.2.37 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP1)—Offset 414h ....3010
65.2.38 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP1)—Offset 418h ....3011
65.2.39 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP1)—Offset 41Ch ....3014
65.2.40 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP2)—Offset 420h ....3015
65.2.41 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP2)—Offset 424h ....3018
65.2.42 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP2)—Offset 428h ....3019
65.2.43 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP2)—Offset 42Ch ....3022
65.2.44 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP3)—Offset 430h ....3023
65.2.45 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP3)—Offset 434h ....3026
65.2.46 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP3)—Offset 438h ....3027

65.2.47 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP3)—Offset 43Ch ..... 3030
65.2.48 Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED0)—Offset 440h ..... 3031
65.2.49 Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED0)—Offset 444h ..... 3034
65.2.50 Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED1)—Offset 448h ..... 3035
65.2.51 Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED1)—Offset 44Ch ..... 3038
65.2.52 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_CLK)—Offset 450h... 3039
65.2.53 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_CLK)—Offset 454h... 3042
65.2.54 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_DATA)—Offset 458h 3043
65.2.55 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_DATA)—Offset 45Ch 3046
65.2.56 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_CLK)—Offset 460h... 3047
65.2.57 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_CLK)—Offset 464h... 3050
65.2.58 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_DATA)—Offset 468h 3051
65.2.59 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_DATA)—Offset 46Ch 3054
65.2.60 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD0)—Offset 470h ..... 3055
65.2.61 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD0)—Offset 474h ..... 3058
65.2.62 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CLK_IN)—Offset 478h..... 3059
65.2.63 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CLK_IN)—Offset 47Ch .... 3062
65.2.64 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD1)—Offset 480h ..... 3063
65.2.65 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD1)—Offset 484h ..... 3066
65.2.66 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CRS_DV)—Offset 488h.... 3067
65.2.67 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CRS_DV)—Offset 48Ch ... 3070
65.2.68 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_IN)—Offset 490h .... 3071
65.2.69 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_IN)—Offset 494h .... 3074
65.2.70 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TX_EN)—Offset 498h..... 3075
65.2.71 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TX_EN)—Offset 49Ch..... 3078
65.2.72 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD0)—Offset 4A0h ..... 3079
65.2.73 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD0)—Offset 4A4h ..... 3082
65.2.74 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD1)—Offset 4A8h ..... 3083
65.2.75 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD1)—Offset 4ACh ..... 3086
65.2.76 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_OUT)—Offset 4B0h.. 3087
65.2.77 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_OUT)—Offset 4B4h.. 3090
65.2.78 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED0)—Offset 4B8h ..... 3091
65.2.79 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED0)—Offset 4BCh ..... 3094
65.2.80 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED1)—Offset 4C0h ..... 3095
65.2.81 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED1)—Offset 4C4h ..... 3098
65.2.82 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED0)—Offset 4C8h ..... 3099
65.2.83 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED0)—Offset 4CCh ..... 3102
65.2.84 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED1)—Offset 4D0h ..... 3103
65.2.85 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED1)—Offset 4D4h ..... 3106
65.2.86 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_0)—Offset 4D8h..... 3107
65.2.87 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_0)—Offset 4DCh..... 3110
65.2.88 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ0_N)—Offset 4E0h ..... 3111
65.2.89 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ0_N)—Offset 4E4h ..... 3114
65.2.90 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ1_N)—Offset 4E8h ..... 3115
65.2.91 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ1_N)—Offset 4ECh ..... 3118
65.2.92 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ2_N)—Offset 4F0h3119
65.2.93 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ2_N)—Offset 4F4h3122
65.2.94 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ3_N)—Offset 4F8h3123
65.2.95 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ3_N)—Offset 4FCh ..... 3126

65.2.96 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ4_N)—Offset 500h ....3127
65.2.97 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ4_N)—Offset 504h ....3130
65.2.98 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_1)—Offset 508h....3131
65.2.99 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_1)—Offset 50Ch....3134
65.2.100 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_2)—Offset 510h....3135
65.2.101 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_2)—Offset 514h....3138
65.2.102 Pad Configuration DW0 (PAD_CFG_DW0_SVID_ALERT_N)—Offset 518h ...3139
65.2.103 Pad Configuration DW1 (PAD_CFG_DW1_SVID_ALERT_N)—Offset 51Ch ...3142
65.2.104 Pad Configuration DW0 (PAD_CFG_DW0_SVID_DATA)—Offset 520h ....3143
65.2.105 Pad Configuration DW1 (PAD_CFG_DW1_SVID_DATA)—Offset 524h ....3146
65.2.106 Pad Configuration DW0 (PAD_CFG_DW0_SVID_CLK)—Offset 528h ....3147
65.2.107 Pad Configuration DW1 (PAD_CFG_DW1_SVID_CLK)—Offset 52Ch ....3150
65.2.108 Pad Configuration DW0 (PAD_CFG_DW0_THERMTRIP_N)—Offset 530h ....3151
65.2.109 Pad Configuration DW1 (PAD_CFG_DW1_THERMTRIP_N)—Offset 534h ....3154
65.2.110 Pad Configuration DW0 (PAD_CFG_DW0_PROCHOT_N)—Offset 538h ....3155
65.2.111 Pad Configuration DW1 (PAD_CFG_DW1_PROCHOT_N)—Offset 53Ch....3158
65.2.112 Pad Configuration DW0 (PAD_CFG_DW0_MEMHOT_N)—Offset 540h ....3159
65.2.113 Pad Configuration DW1 (PAD_CFG_DW1_MEMHOT_N)—Offset 544h ....3162
65.3 Sideband Registers, Part 2 of 2—Sideband Port 0xC5 ....3163
65.3.1 Pad Base Address (PADBAR)—Offset 0Ch ....3163
65.3.2 Miscellaneous Configuration (MISCCFG)—Offset 10h ....3164
65.3.3 Pad Ownership (PAD_OWN_SOUTH_DFX_0)—Offset 20h ....3165
65.3.4 Pad Ownership (PAD_OWN_SOUTH_DFX_1)—Offset 24h ....3167
65.3.5 Pad Ownership (PAD_OWN_SOUTH_DFX_2)—Offset 28h ....3169
65.3.6 Pad Ownership (PAD_OWN_SOUTH_GROUP0_0)—Offset 2Ch....3170
65.3.7 Pad Ownership (PAD_OWN_SOUTH_GROUP0_1)—Offset 30h ....3172
65.3.8 Pad Ownership (PAD_OWN_SOUTH_GROUP0_2)—Offset 34h ....3174
65.3.9 Pad Ownership (PAD_OWN_SOUTH_GROUP0_3)—Offset 38h ....3176
65.3.10 Pad Ownership (PAD_OWN_SOUTH_GROUP0_4)—Offset 3Ch....3178
65.3.11 Pad Ownership (PAD_OWN_SOUTH_GROUP0_5)—Offset 40h ....3180
65.3.12 Pad Ownership (PAD_OWN_SOUTH_GROUP0_6)—Offset 44h ....3182
65.3.13 Pad Ownership (PAD_OWN_SOUTH_GROUP1_0)—Offset 48h ....3183
65.3.14 Pad Ownership (PAD_OWN_SOUTH_GROUP1_1)—Offset 4Ch....3185
65.3.15 Pad Ownership (PAD_OWN_SOUTH_GROUP1_2)—Offset 50h ....3187
65.3.16 Pad Ownership (PAD_OWN_SOUTH_GROUP1_3)—Offset 54h ....3189
65.3.17 Pad Ownership (PAD_OWN_SOUTH_GROUP1_4)—Offset 58h ....3191
65.3.18 Pad Ownership (PAD_OWN_SOUTH_GROUP1_5)—Offset 5Ch....3193
65.3.19 Pad Configuration Lock (PADCFGLOCK_SOUTH_DFX_0)—Offset 90h ....3194
65.3.20 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_DFX_0)—Offset 94h....3196
65.3.21 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_0)—Offset 98h ..3198
65.3.22 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_0)—Offset 9Ch ....3200
65.3.23 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_1)—Offset A0h ..3202
65.3.24 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_1)—Offset A4h....3204
65.3.25 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_0)—Offset A8h ..3206
65.3.26 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_0)—Offset ACh ....3208
65.3.27 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_1)—Offset B0h ..3210
65.3.28 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_1)—Offset B4h ....3211
65.3.29 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_DFX_0)—Offset C0h ....3212

65.3.30 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_0)—Offset C4h ..... 3214
65.3.31 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_1)—Offset C8h ..... 3216
65.3.32 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_0)—Offset CCh ..... 3218
65.3.33 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_1)—Offset D0h ..... 3220
65.3.34 GPI Interrupt Status (GPI_IS_SOUTH_DFX_0)—Offset 100h..... 3221
65.3.35 GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_0)—Offset 104h..... 3223
65.3.36 GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_1)—Offset 108h..... 3225
65.3.37 GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_0)—Offset 10Ch ..... 3227
65.3.38 GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_1)—Offset 110h..... 3229
65.3.39 GPI Interrupt Enable (GPI_IE_SOUTH_DFX_0)—Offset 120h ..... 3230
65.3.40 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_0)—Offset 124h ..... 3232
65.3.41 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_1)—Offset 128h ..... 3234
65.3.42 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_0)—Offset 12Ch ..... 3236
65.3.43 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_1)—Offset 130h ..... 3238
65.3.44 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_DFX_0)—Offset 140h ..... 3239
65.3.45 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_0)—Offset 144h ..... 3241
65.3.46 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_1)—Offset 148h ..... 3243
65.3.47 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_0)—Offset 14Ch ..... 3245
65.3.48 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_1)—Offset 150h ..... 3247
65.3.49 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_DFX_0)—Offset 160h..... 3248
65.3.50 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_0)—Offset 164h..... 3250
65.3.51 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_1)—Offset 168h..... 3252
65.3.52 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_0)—Offset 16Ch..... 3254
65.3.53 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_1)—Offset 170h..... 3256
65.3.54 SMI Status (GPI_SMI_STS_SOUTH_GROUP0_0)—Offset 184h ..... 3257
65.3.55 SMI Status (GPI_SMI_STS_SOUTH_GROUP0_1)—Offset 188h ..... 3259
65.3.56 SMI Status (GPI_SMI_STS_SOUTH_GROUP1_0)—Offset 18Ch ..... 3261
65.3.57 SMI Status (GPI_SMI_STS_SOUTH_GROUP1_1)—Offset 190h ..... 3263
65.3.58 SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_0)—Offset 1A4h ..... 3264
65.3.59 SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_1)—Offset 1A8h ..... 3266
65.3.60 SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_0)—Offset 1ACh ..... 3268
65.3.61 SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_1)—Offset 1B0h ..... 3270
65.3.62 NMI Status (GPI_NMI_STS_SOUTH_GROUP0_0)—Offset 1C4h..... 3271
65.3.63 NMI Status (GPI_NMI_STS_SOUTH_GROUP0_1)—Offset 1C8h..... 3273
65.3.64 NMI Status (GPI_NMI_STS_SOUTH_GROUP1_0)—Offset 1CCh ..... 3275
65.3.65 NMI Status (GPI_NMI_STS_SOUTH_GROUP1_1)—Offset 1D0h ..... 3277
65.3.66 NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_0)—Offset 1E4h ..... 3278
65.3.67 NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_1)—Offset 1E8h ..... 3280
65.3.68 NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_0)—Offset 1ECh..... 3282
65.3.69 NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_1)—Offset 1F0h ..... 3284
65.3.70 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK0)—Offset 400h 3285
65.3.71 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK0)—Offset 404h 3288

65.3.72 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK1)—Offset 408h.3289
65.3.73 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK1)—Offset 40Ch.3292
65.3.74 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT0)—Offset 410h.....3293
65.3.75 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT0)—Offset 414h.....3296
65.3.76 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT1)—Offset 418h.....3297
65.3.77 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT1)—Offset 41Ch.....3300
65.3.78 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT2)—Offset 420h.....3301
65.3.79 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT2)—Offset 424h.....3304
65.3.80 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT3)—Offset 428h.....3305
65.3.81 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT3)—Offset 42Ch.....3308
65.3.82 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT4)—Offset 430h.....3309
65.3.83 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT4)—Offset 434h.....3312
65.3.84 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT5)—Offset 438h.....3313
65.3.85 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT5)—Offset 43Ch.....3316
65.3.86 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT6)—Offset 440h.....3317
65.3.87 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT6)—Offset 444h.....3320
65.3.88 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT7)—Offset 448h.....3321
65.3.89 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT7)—Offset 44Ch.....3324
65.3.90 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT8)—Offset 450h.....3325
65.3.91 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT8)—Offset 454h.....3328
65.3.92 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT9)—Offset 458h.....3329
65.3.93 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT9)—Offset 45Ch.....3332
65.3.94 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT10)—Offset 460h .....3333
65.3.95 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT10)—Offset 464h .....3336
65.3.96 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT11)—Offset 468h .....3337
65.3.97 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT11)—Offset 46Ch .....3340
65.3.98 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT12)—Offset 470h .....3341
65.3.99 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT12)—Offset 474h .....3344
65.3.100 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT13)—Offset 478h .....3345
65.3.101 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT13)—Offset 47Ch .....3348
65.3.102 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT14)—Offset 480h .....3349
65.3.103 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT14)—Offset 484h .....3352
65.3.104 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT15)—Offset 488h .....3353
65.3.105 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT15)—Offset 48Ch .....3356
65.3.106 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_12)—Offset 490h .....3357
65.3.107 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_12)—Offset 494h .....3360
65.3.108 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_ALRT_N)—Offset 498h.....3361
65.3.109 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_ALRT_N)—Offset 49Ch.....3364
65.3.110 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ5_N)—Offset 4A0h.....3365
65.3.111 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ5_N)—Offset 4A4h.....3368
65.3.112 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ6_N)—Offset 4A8h.....3369
65.3.113 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ6_N)—Offset 4ACh.....3372
65.3.114 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ7_N)—Offset 4B0h.....3373
65.3.115 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ7_N)—Offset 4B4h.....3376
65.3.116 Pad Configuration DW0 (PAD_CFG_DW0_UART0_RXD)—Offset 4B8h .....3377
65.3.117 Pad Configuration DW1 (PAD_CFG_DW1_UART0_RXD)—Offset 4BCh .....3380
65.3.118 Pad Configuration DW0 (PAD_CFG_DW0_UART0_TXD)—Offset 4C0h .....3381

65.3.119 Pad Configuration DW1 (PAD_CFG_DW1_UART0_TXD)—Offset 4C4h ..... 3384
65.3.120 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_CLK)—Offset 4C8h . 3385
65.3.121 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_CLK)—Offset 4CCh . 3388
65.3.122 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_DATA)—Offset 4D0h ..... 3389
65.3.123 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_DATA)—Offset 4D4h ..... 3392
65.3.124 Pad Configuration DW0 (PAD_CFG_DW0_ERROR2_N)—Offset 4D8h ..... 3393
65.3.125 Pad Configuration DW1 (PAD_CFG_DW1_ERROR2_N)—Offset 4DCh ..... 3396
65.3.126 Pad Configuration DW0 (PAD_CFG_DW0_ERROR1_N)—Offset 4E0h..... 3397
65.3.127 Pad Configuration DW1 (PAD_CFG_DW1_ERROR1_N)—Offset 4E4h..... 3400
65.3.128 Pad Configuration DW0 (PAD_CFG_DW0_ERROR0_N)—Offset 4E8h..... 3401
65.3.129 Pad Configuration DW1 (PAD_CFG_DW1_ERROR0_N)—Offset 4ECh ..... 3404
65.3.130 Pad Configuration DW0 (PAD_CFG_DW0_IERR_N)—Offset 4F0h ..... 3405
65.3.131 Pad Configuration DW1 (PAD_CFG_DW1_IERR_N)—Offset 4F4h ..... 3408
65.3.132 Pad Configuration DW0 (PAD_CFG_DW0_MCERR_N)—Offset 4F8h ..... 3409
65.3.133 Pad Configuration DW1 (PAD_CFG_DW1_MCERR_N)—Offset 4FCh ..... 3412
65.3.134 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_CLK)—Offset 500h.. 3413
65.3.135 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_CLK)—Offset 504h.. 3416
65.3.136 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_DATA)—Offset 508h 3417
65.3.137 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_DATA)—Offset 50Ch 3420
65.3.138 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_ALRT_N)—Offset 510h ..... 3421
65.3.139 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_ALRT_N)—Offset 514h ..... 3424
65.3.140 Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_DATA)—Offset 518h ..... 3425
65.3.141 Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_DATA)—Offset 51Ch ..... 3428
65.3.142 Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_CLK)—Offset 520h 3429
65.3.143 Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_CLK)—Offset 524h 3432
65.3.144 Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_DATA)—Offset 528h ..... 3433
65.3.145 Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_DATA)—Offset 52Ch ..... 3436
65.3.146 Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_CLK)—Offset 530h. 3437
65.3.147 Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_CLK)—Offset 534h. 3440
65.3.148 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_DATA)—Offset 538h ..... 3441
65.3.149 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_DATA)—Offset 53Ch ..... 3444
65.3.150 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_CLK)—Offset 540h ..... 3445
65.3.151 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_CLK)—Offset 544h ..... 3448
65.3.152 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_ALRT_N)—Offset 548h ..... 3449
65.3.153 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_ALRT_N)—Offset 54Ch ..... 3452
65.3.154 Pad Configuration DW0 (PAD_CFG_DW0_USB_OC0_N)—Offset 550h..... 3453
65.3.155 Pad Configuration DW1 (PAD_CFG_DW1_USB_OC0_N)—Offset 554h..... 3456
65.3.156 Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE0)—Offset 558h... 3457
65.3.157 Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE0)—Offset 55Ch... 3460
65.3.158 Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE1)—Offset 560h... 3461
65.3.159 Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE1)—Offset 564h... 3464
65.3.160 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_4)—Offset 568h ..... 3465

65.3.161 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_4)—Offset 56Ch.....3468
65.3.162 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_5)—Offset 570h.....3469
65.3.163 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_5)—Offset 574h.....3472
65.3.164 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_6)—Offset 578h.....3473
65.3.165 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_6)—Offset 57Ch.....3476
65.3.166 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_7)—Offset 580h.....3477
65.3.167 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_7)—Offset 584h.....3480
65.3.168 Pad Configuration DW0 (PAD_CFG_DW0_SATA0_LED_N)—Offset 588h ....3481
65.3.169 Pad Configuration DW1 (PAD_CFG_DW1_SATA0_LED_N)—Offset 58Ch ....3484
65.3.170 Pad Configuration DW0 (PAD_CFG_DW0_SATA1_LED_N)—Offset 590h ....3485
65.3.171 Pad Configuration DW1 (PAD_CFG_DW1_SATA1_LED_N)—Offset 594h ....3488
65.3.172 Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT0)—Offset 598h.3489
65.3.173 Pad Configuration DW1 (PAD_CFG_DW1_SATA_PDETECT0)—Offset 59Ch.3492
65.3.174 Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT1)—Offset 5A0h.3493
65.3.175 Pad Configuration DW1 (PAD_CFG_DW1)—Offset 5A4h.....3496
65.3.176 Pad Configuration DW0 (PAD_CFG_DW0_SATA0_SDOUT)—Offset 5A8h ...3497
65.3.177 Pad Configuration DW1 (PAD_CFG_DW1_SATA0_SDOUT)—Offset 5ACh ...3500
65.3.178 Pad Configuration DW0 (PAD_CFG_DW0_SATA1_SDOUT)—Offset 5B0h ...3501
65.3.179 Pad Configuration DW1 (PAD_CFG_DW1_SATA1_SDOUT)—Offset 5B4h ...3504
65.3.180 Pad Configuration DW0 (PAD_CFG_DW0_UART1_RXD)—Offset 5B8h .....3505
65.3.181 Pad Configuration DW1 (PAD_CFG_DW1_UART1_RXD)—Offset 5BCh .....3508
65.3.182 Pad Configuration DW0 (PAD_CFG_DW0_UART1_TXD)—Offset 5C0h .....3509
65.3.183 Pad Configuration DW1 (PAD_CFG_DW1_UART1_TXD)—Offset 5C4h .....3512
65.3.184 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_8)—Offset 5C8h.....3513
65.3.185 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_8)—Offset 5CCh .....3516
65.3.186 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_9)—Offset 5D0h .....3517
65.3.187 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_9)—Offset 5D4h .....3520
65.3.188 Pad Configuration DW0 (PAD_CFG_DW0_TCK)—Offset 5D8h .....3521
65.3.189 Pad Configuration DW1 (PAD_CFG_DW1_TCK)—Offset 5DCh .....3524
65.3.190 Pad Configuration DW0 (PAD_CFG_DW0_TRST_N)—Offset 5E0h .....3525
65.3.191 Pad Configuration DW1 (PAD_CFG_DW1_TRST_N)—Offset 5E4h .....3528
65.3.192 Pad Configuration DW0 (PAD_CFG_DW0_TMS)—Offset 5E8h .....3529
65.3.193 Pad Configuration DW1 (PAD_CFG_DW1_TMS)—Offset 5ECh .....3532
65.3.194 Pad Configuration DW0 (PAD_CFG_DW0_TDI)—Offset 5F0h .....3533
65.3.195 Pad Configuration DW1 (PAD_CFG_DW1_TDI)—Offset 5F4h .....3536
65.3.196 Pad Configuration DW0 (PAD_CFG_DW0_TDO)—Offset 5F8h .....3537
65.3.197 Pad Configuration DW1 (PAD_CFG_DW1_TDO)—Offset 5FCh .....3540
65.3.198 Pad Configuration DW0 (PAD_CFG_DW0_CX_PRDY_N)—Offset 600h.....3541
65.3.199 Pad Configuration DW1 (PAD_CFG_DW1_CX_PRDY_N)—Offset 604h.....3544
65.3.200 Pad Configuration DW0 (PAD_CFG_DW0_CX_PREQ_N)—Offset 608h .....3545
65.3.201 Pad Configuration DW1 (PAD_CFG_DW1_CX_PREQ_N)—Offset 60Ch .....3548
65.3.202 Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGINOUT)—Offset 610h...3549
65.3.203 Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGINOUT)—Offset 614h...3552
65.3.204 Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGOUT)—Offset 618h .....3553
65.3.205 Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGOUT)—Offset 61Ch .....3556
65.3.206 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE2)—Offset 620h .....3557
65.3.207 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE2)—Offset 624h .....3560
65.3.208 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE3)—Offset 628h .....3561
65.3.209 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE3)—Offset 62Ch .....3564
65.3.210 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE4)—Offset 630h .....3565
65.3.211 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE4)—Offset 634h .....3568
65.3.212 Pad Configuration DW0 (PAD_CFG_DW0_SUSPWRDNACK)—Offset 638h ..3569
65.3.213 Pad Configuration DW1 (PAD_CFG_DW1_SUSPWRDNACK)—Offset 63Ch ..3572
65.3.214 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SUSCLK)—Offset 640h .....3573
65.3.215 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SUSCLK)—Offset 644h .....3576

65.3.216 Pad Configuration DW0 (PAD_CFG_DW0_ADR_TRIGGER)—Offset 648h ... 3577
65.3.217 Pad Configuration DW1 (PAD_CFG_DW1_ADR_TRIGGER)—Offset 64Ch ... 3580
65.3.218 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S45_N)—Offset 650h 3581
65.3.219 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S45_N)—Offset 654h 3584
65.3.220 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S3_N)—Offset 658h.. 3585
65.3.221 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S3_N)—Offset 65Ch.. 3588
65.3.222 Pad Configuration DW0 (PAD_CFG_DW0_PMU_WAKE_N)—Offset 660h.... 3589
65.3.223 Pad Configuration DW1 (PAD_CFG_DW1_PMU_WAKE_N)—Offset 664h.... 3592
65.3.224 Pad Configuration DW0 (PAD_CFG_DW0_PMU_PWRBTN_N)—Offset 668h 3593
65.3.225 Pad Configuration DW1 (PAD_CFG_DW1_PMU_PWRBTN_N)—Offset 66Ch 3596
65.3.226 Pad Configuration DW0 (PAD_CFG_DW0_PMU_RESETBUTTON_N)—Offset 670h ..... 3597
65.3.227 Pad Configuration DW1 (PAD_CFG_DW1_PMU_RESETBUTTON_N)—Offset 674h ..... 3600
65.3.228 Pad Configuration DW0 (PAD_CFG_DW0_PMU_PLTRST_N)—Offset 678h.. 3601
65.3.229 Pad Configuration DW1 (PAD_CFG_DW1_PMU_PLTRST_N)—Offset 67Ch.. 3604
65.3.230 Pad Configuration DW0 (PAD_CFG_DW0_SUS_STAT_N)—Offset 680h ..... 3605
65.3.231 Pad Configuration DW1 (PAD_CFG_DW1_SUS_STAT_N)—Offset 684h ..... 3608
65.3.232 Pad Configuration DW0 (PAD_CFG_DW0_SLP_S0IX_N)—Offset 688h ..... 3609
65.3.233 Pad Configuration DW1 (PAD_CFG_DW1_SLP_S0IX_N)—Offset 68Ch ..... 3612
65.3.234 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS0_N)—Offset 690h ..... 3613
65.3.235 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS0_N)—Offset 694h ..... 3616
65.3.236 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS1_N)—Offset 698h ..... 3617
65.3.237 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS1_N)—Offset 69Ch..... 3620
65.3.238 Pad Configuration DW0 (PAD_CFG_DW0_SPI_MOSI_IO0)—Offset 6A0h... 3621
65.3.239 Pad Configuration DW1 (PAD_CFG_DW1_SPI_MOSI_IO0)—Offset 6A4h... 3624
65.3.240 Pad Configuration DW0 (PAD_CFG_DW0_SPI_MISO_IO1)—Offset 6A8h... 3625
65.3.241 Pad Configuration DW1 (PAD_CFG_DW1_SPI_MISO_IO1)—Offset 6ACh... 3628
65.3.242 Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO2)—Offset 6B0h ..... 3629
65.3.243 Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO2)—Offset 6B4h ..... 3632
65.3.244 Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO3)—Offset 6B8h ..... 3633
65.3.245 Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO3)—Offset 6BCh ..... 3636
65.3.246 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK)—Offset 6C0h..... 3637
65.3.247 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK)—Offset 6C4h..... 3640
65.3.248 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK_LOOPBK)—Offset 6C8h ..... 3641
65.3.249 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK_LOOPBK)—Offset 6CCh ..... 3644
65.3.250 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO0)—Offset 6D0h ..... 3645
65.3.251 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO0)—Offset 6D4h ..... 3648
65.3.252 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO1)—Offset 6D8h ..... 3649
65.3.253 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO1)—Offset 6DCh ..... 3652
65.3.254 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO2)—Offset 6E0h ..... 3653
65.3.255 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO2)—Offset 6E4h ..... 3656
65.3.256 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO3)—Offset 6E8h ..... 3657
65.3.257 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO3)—Offset 6ECh ..... 3660
65.3.258 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CS0_N)—Offset 6F0h ..... 3661
65.3.259 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CS0_N)—Offset 6F4h ..... 3664
65.3.260 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK)—Offset 6F8h ..... 3665
65.3.261 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK)—Offset 6FCh ..... 3668
65.3.262 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_RST_N)—Offset 700h ..... 3669
65.3.263 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_RST_N)—Offset 704h ..... 3672
65.3.264 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_ALRT0_N)—Offset 708h... 3673
65.3.265 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_ALRT0_N)—Offset 70Ch... 3676
65.3.266 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_10)—Offset 710h ..... 3677

65.3.267 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_10)—Offset 714h .....3680
65.3.268 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_11)—Offset 718h .....3681
65.3.269 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_11)—Offset 71Ch .....3684
65.3.270 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK_LOOPBK)—Offset 720h .....3685
65.3.271 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK_LOOPBK)—Offset 724h .....3688
65.3.272 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CMD)—Offset 728h .....3689
65.3.273 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CMD)—Offset 72Ch .....3692
65.3.274 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_STROBE)—Offset 730h ...3693
65.3.275 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_STROBE)—Offset 734h ...3696
65.3.276 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CLK)—Offset 738h .....3697
65.3.277 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CLK)—Offset 73Ch .....3700
65.3.278 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D0)—Offset 740h .....3701
65.3.279 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D0)—Offset 744h .....3704
65.3.280 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D1)—Offset 748h .....3705
65.3.281 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D1)—Offset 74Ch .....3708
65.3.282 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D2)—Offset 750h .....3709
65.3.283 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D2)—Offset 754h .....3712
65.3.284 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D3)—Offset 758h .....3713
65.3.285 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D3)—Offset 75Ch .....3716
65.3.286 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D4)—Offset 760h .....3717
65.3.287 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D4)—Offset 764h .....3720
65.3.288 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D5)—Offset 768h .....3721
65.3.289 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D5)—Offset 76Ch .....3724
65.3.290 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D6)—Offset 770h .....3725
65.3.291 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D6)—Offset 774h .....3728
65.3.292 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D7)—Offset 778h .....3729
65.3.293 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D7)—Offset 77Ch .....3732
65.3.294 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_3)—Offset 780h.....3733
65.3.295 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_3)—Offset 784h.....3736

**66 SMBus - PECI ..... 3737**
66.1 Introduction and Index .....3737

**67 Interrupt Steering and Control..... 3738**
67.1 Introduction and Index .....3738
67.1.1 Sideband Registers .....3739
67.2 Sideband Registers.....3741
67.2.1 PIRQA Routing Control (PARC)—Offset 3100h .....3741
67.2.2 PIRQB Routing Control (PBRC)—Offset 3101h .....3742
67.2.3 PIRQC Routing Control (PCRC)—Offset 3102h .....3743
67.2.4 PIRQD Routing Control (PDRC)—Offset 3103h.....3744
67.2.5 PIRQE Routing Control (PERC)—Offset 3104h .....3745
67.2.6 PIRQF Routing Control (PFRC)—Offset 3105h.....3746
67.2.7 PIRQG Routing Control (PGRC)—Offset 3106h.....3747
67.2.8 PIRQH Routing Control (PHRC)—Offset 3107h.....3748
67.2.9 Message Decoder Control (MSGDC)—Offset 3120h .....3749
67.2.10 PCI Interrupt Route 0 (PIR0)—Offset 3140h .....3750
67.2.11 PCI Interrupt Route 1 (PIR1)—Offset 3142h .....3751
67.2.12 PCI Interrupt Route 2 (PIR2)—Offset 3144h .....3752
67.2.13 PCI Interrupt Route 3 (PIR3)—Offset 3146h .....3753
67.2.14 PCI Interrupt Route 4 (PIR4)—Offset 3148h .....3754
67.2.15 PCI Interrupt Route 5 (PIR5)—Offset 314Ah .....3755
67.2.16 PCI Interrupt Route 6 (PIR6)—Offset 314Ch .....3756
67.2.17 PCI Interrupt Route 7 (PIR7)—Offset 314Eh .....3757

67.2.18 PCI Interrupt Route 8 (PIR8)—Offset 3150h ..... 3758
67.2.19 PCI Interrupt Route 9 (PIR9)—Offset 3152h ..... 3759
67.2.20 PCI Interrupt Route 10 (PIR10)—Offset 3154h ..... 3760
67.2.21 PCI Interrupt Route 11 (PIR11)—Offset 3156h ..... 3761
67.2.22 PCI Interrupt Route 12 (PIR12)—Offset 3158h ..... 3762
67.2.23 General Interrupt Control (GIC)—Offset 31FCh ..... 3763
67.2.24 Interrupt Polarity Control 0 (IPC0)—Offset 3200h ..... 3765
67.2.25 Interrupt Polarity Control 1 (IPC1)—Offset 3204h ..... 3765
67.2.26 Interrupt Polarity Control 2 (IPC2)—Offset 3208h ..... 3765
67.2.27 Interrupt Polarity Control 3 (IPC3)—Offset 320Ch ..... 3766
67.2.28 Interrupt Blocking Control (IBC)—Offset 3220h ..... 3766
67.2.29 Interrupt Edge-Trigger Extension 0 (IETE0)—Offset 3230h ..... 3766
67.2.30 Interrupt Edge-Trigger Extension 1 (IETE1)—Offset 3234h ..... 3767
67.2.31 Interrupt Edge-Trigger Extension 2 (IETE2)—Offset 3238h ..... 3767
67.2.32 Interrupt Edge-Trigger Extension 3 (IETE3)—Offset 323Ch ..... 3767
67.2.33 ITSS Power Reduction Control (ITSSPRC)—Offset 3300h ..... 3768
67.2.34 SIDE Clock Timing (SIDECT)—Offset 3304h ..... 3769
67.2.35 IPCI Clock Timing (IPCICT)—Offset 3306h ..... 3770
67.2.36 PGCB Clock Timing (PGCBCT)—Offset 3308h ..... 3770
67.2.37 Uncorrectable Error Mask (UEM)—Offset 3320h ..... 3771
67.2.38 Uncorrectable Error Severity (UEV)—Offset 3324h ..... 3772
67.2.39 Correctable Error Mask (CEM)—Offset 3328h ..... 3773
67.2.40 NMI Control (NMI)—Offset 3330h ..... 3774
67.2.41 Master Message Control (MMC)—Offset 3334h ..... 3775
67.2.42 Master Message Status (MMSTS)—Offset 3336h ..... 3775
67.2.43 HPET Offload Scale Value (HOFFVAL)—Offset 3400h ..... 3776

**68 PHY - High-Speed I/O ..... 3777**
68.1 Introduction and Index ..... 3777
68.1.1 Sideband Registers ..... 3778
68.2 Sideband Registers ..... 3788
68.2.1 PCS_DWORD10 (pcs_dword10)—Offset 28h ..... 3788
68.2.2 TX_DWORD5 (tx_dword5)—Offset 94h ..... 3789
68.2.3 TX_DWORD6 (tx_dword6)—Offset 98h ..... 3790
68.2.4 TX_DWORD7 (tx_dword7)—Offset 9Ch ..... 3791
68.2.5 TX_DWORD19 (tx_dword19)—Offset CCh ..... 3792
68.2.6 RX_DWORD20 (rx_dword20)—Offset 150h ..... 3792
68.2.7 RX_DWORD21 (rx_dword21)—Offset 154h ..... 3793
68.2.8 RX_DWORD25 (rx_dword25)—Offset 164h ..... 3794
68.2.9 RX_DWORD26 (rx_dword26)—Offset 168h ..... 3794

**69 Model-Specific Registers (MSR) ..... 3795**
69.1 Introduction and Index ..... 3795
69.2 Model Specific Registers (MSRs) ..... 3801
69.2.1 (006h) IA32_MONITOR_FILTER_SIZE ..... 3801
69.2.2 (017h) IA32_PLATFORM_ID ..... 3802
69.2.3 (01Bh) IA32_APIC_BASE ..... 3802
69.2.4 (02Eh) MSR_PIC_MSG_CONTROL ..... 3803
69.2.5 (033h) MSR_MEMORY_CONTROL ..... 3803
69.2.6 (034h) MSR_SMI_COUNT ..... 3803
69.2.7 (035h) CORE_THREAD_COUNT ..... 3804
69.2.8 (039h) MSR_THREAD_ID ..... 3804
69.2.9 (03Bh) IA32_TSC_ADJUST ..... 3804
69.2.10 (04Eh) MSR_PPIN_CTL ..... 3805
69.2.11 (04Fh) MSR_PPIN ..... 3805
69.2.12 (052h) SMM_MCA_CONTROL ..... 3805

69.2.13 (079h) IA32_BIOS_UPDT_TRIG ....3806
69.2.14 (07Ah) MSR_SGX_FEATURE_ACTIVATION ....3806
69.2.15 (08Bh) IA32_BIOS_SIGN_ID ....3806
69.2.16 (09Bh) IA32_SMM_MONITOR_CTL ....3806
69.2.17 (09Eh) IA32_SMBASE....3807
69.2.18 (0C1h) IA32_PMC0 ....3807
69.2.19 (0C2h) IA32_PMC1 ....3807
69.2.20 (0C3h) IA32_PMC2 ....3807
69.2.21 (0C4h) IA32_PMC3 ....3807
69.2.22 (0CEh) MSR_PLATFORM_INFO ....3808
69.2.23 (0E2h) MSR_PKG_CST_CONFIG_CONTROL....3810
69.2.24 (0E4h) MSR_PMG_IO_CAPTURE_BASE....3811
69.2.25 (0E7h) IA32_MPERF ....3811
69.2.26 (0E8h) IA32_APERF ....3811
69.2.27 (0FEh) IA32_MTRRCAP ....3812
69.2.28 (120h) POWER_MISC ....3812
69.2.29 (121h) EMULATE_PM_TMR ....3813
69.2.30 (139h) INITIAL_BOOT_BLOCK_COMPLETE....3813
69.2.31 (13Ch) FEATURE_CONFIG ....3813
69.2.32 (140h) FEATURE_ENABLES ....3814
69.2.33 (15Ch) PERMEM_CONFIG_INFO ....3814
69.2.34 (15Dh) PERMEM_CTRLR_INDEX ....3814
69.2.35 (15Eh) PERMEM_CTRLR_ID ....3814
69.2.36 (174h) IA32_SYSENTER_CS ....3815
69.2.37 (175h) IA32_SYSENTER_ESP....3815
69.2.38 (176h) IA32_SYSENTER_EIP ....3815
69.2.39 (179h) IA32_MCG_CAP....3816
69.2.40 (17Ah) IA32_MCG_STATUS ....3816
69.2.41 (17Dh) MSR_SMM_MCA_CAP....3817
69.2.42 (17Fh) MSR_ERROR_CONTROL ....3817
69.2.43 (186h) IA32_PERFEVTSEL0 ....3818
69.2.44 (187h) IA32_PERFEVTSEL1 ....3818
69.2.45 (188h) IA32_PERFEVTSEL2 ....3818
69.2.46 (189h) IA32_PERFEVTSEL3 ....3819
69.2.47 (198h) IA32_PERF_STATUS ....3819
69.2.48 (199h) IA32_PERF_CTL ....3819
69.2.49 (19Ah) IA32_CLOCK_MODULATION....3820
69.2.50 (19Bh) IA32_THERM_INTERRUPT....3820
69.2.51 (19Ch) IA32_THERM_STATUS ....3821
69.2.52 (19Dh) MSR_THERM2_CTL....3822
69.2.53 (1A0h) IA32_MISC_ENABLE ....3823
69.2.54 (1A1h) PACKAGE_THERM_MARGINBD....3824
69.2.55 (1A2h) MSR_TEMPERATURE_TARGET ....3824
69.2.56 (1A4h) MSR_MISC_FEATURE_CONTROL....3824
69.2.57 (1A6h) MSR_OFFCORE_RSP_0....3825
69.2.58 (1A7h) MSR_OFFCORE_RSP_1....3825
69.2.59 (1A8h) THREAD_SW_DEFEATURE ....3825
69.2.60 (1AAh) MSR_MISC_PWR_MGMT....3825
69.2.61 (1ADh) MSR_TURBO_RATIO_LIMIT ....3826
69.2.62 (1AEh) MSR_TURBO_RATIO_LIMIT1 ....3826
69.2.63 (1B0h) IA32_ENERGY_PERF_BIAS....3826
69.2.64 (1B1h) IA32_PACKAGE_THERM_STATUS....3827
69.2.65 (1B2h) IA32_PACKAGE_THERM_INTERRUPT ....3828
69.2.66 (1C6h) DEBUG_RESOURCE_STATUS....3828
69.2.67 (1C8h) MSR_LBR_SELECT ....3829

69.2.68 (1C9h) MSR_LASTBRANCH_TOS ..... 3829
69.2.69 (1D9h) IA32_DEBUGCTL ..... 3830
69.2.70 (1E3h) DEBUG_ERR_INJ_CTL ..... 3830
69.2.71 (1E4h) DEBUG_ERR_INJ_CTL2 ..... 3831
69.2.72 (1E6h) PIC_DEBUG_MODES ..... 3832
69.2.73 (1F0h) VLW_CAPABILITY ..... 3834
69.2.74 (1FCh) MSR_POWER_CTL ..... 3835
69.2.75 (281h) IA32_MC1_CTL2 (L2 Cache) ..... 3836
69.2.76 (282h) IA32_MC2_CTL2 (FEC) ..... 3836
69.2.77 (283h) IA32_MC3_CTL2 (MEC) ..... 3836
69.2.78 (284h) IA32_MC4_CTL2 (SA) ..... 3836
69.2.79 (285h) IA32_MC5_CTL2 (PMU) ..... 3837
69.2.80 (286h) IA32_MC6_CTL2 (SA) ..... 3837
69.2.81 (287h) IA32_MC7_CTL2 (MEM 0) ..... 3837
69.2.82 (288h) IA32_MC8_CTL2 (MEM 1) ..... 3838
69.2.83 (2E0h) NO_EVICT_MODE ..... 3838
69.2.84 (2FFh) IA32_MTRR_DEF_TYPE ..... 3838
69.2.85 (300h) MSR_SGXOWNER0 ..... 3839
69.2.86 (301h) MSR_SGXOWNER1 ..... 3839
69.2.87 (309h) IA32_FIXED_CTR0 ..... 3839
69.2.88 (30Ah) IA32_FIXED_CTR1 ..... 3839
69.2.89 (30Bh) IA32_FIXED_CTR2 ..... 3839
69.2.90 (345h) IA32_PERF_CAPABILITIES ..... 3840
69.2.91 (38Dh) IA32_FIXED_CTR_CTRL ..... 3840
69.2.92 (38Eh) IA32_PERF_GLOBAL_STATUS ..... 3841
69.2.93 (38Fh) IA32_PERF_GLOBAL_CTRL ..... 3842
69.2.94 (390h) IA32_PERF_GLOBAL_STATUS_RESET ..... 3843
69.2.95 (391h) IA32_PERF_GLOBAL_STATUS_SET ..... 3844
69.2.96 (392h) IA32_PERF_GLOBAL_INUSE ..... 3844
69.2.97 (3F1h) IA32_PEBS_ENABLE ..... 3845
69.2.98 (3F8h) MSR_PKG_C3_RESIDENCY ..... 3845
69.2.99 (3F9h) MSR_PKG_C6_RESIDENCY ..... 3845
69.2.100 (3FCh) MSR_CORE_C3_RESIDENCY ..... 3845
69.2.101 (3FDh) MSR_CORE_C6_RESIDENCY ..... 3846
69.2.102 (3FEh) MSR_CORE_C7_RESIDENCY ..... 3846
69.2.103 (400h) IA32_MC0_CTL (BIU) ..... 3846
69.2.104 (401h) IA32_MC0_STATUS ..... 3847
69.2.105 (402h) IA32_MC0_ADDR (BIU) ..... 3848
69.2.106 (404h) IA32_MC1_CTL (L2 Cache) ..... 3848
69.2.107 (405h) IA32_MC1_STATUS (L2 Cache) ..... 3849
69.2.108 (406h) IA32_MC1_ADDR (L2 Cache) ..... 3851
69.2.109 (408h) IA32_MC2_CTL (FEC) ..... 3851
69.2.110 (409h) IA32_MC2_STATUS (FEC) ..... 3852
69.2.111 (40Ah) IA32_MC2_ADDR (FEC) ..... 3854
69.2.112 (40Ch) IA32_MC3_CTL (MEC) ..... 3854
69.2.113 (40Dh) IA32_MC3_STATUS (MEC) ..... 3855
69.2.114 (40Eh) IA32_MC3_ADDR (MEC) ..... 3855
69.2.115 (410h) IA32_MC4_CTL (SA) ..... 3856
69.2.116 (411h) IA32_MC4_STATUS (SA) ..... 3858
69.2.117 (412h) IA32_MC4_ADDR (SA) ..... 3860
69.2.118 (413h) IA32_MC4_MISC (SA) ..... 3860
69.2.119 (414h) IA32_MC5_CTL (PMU) ..... 3861
69.2.120 (415h) IA32_MC5_STATUS (PMU) ..... 3863
69.2.121 (416h) IA32_MC5_ADDR (PMU) ..... 3864
69.2.122 (417h) IA32_MC5_MISC (PMU) ..... 3864

69.2.123 (418h) IA32_MC6_CTL (SA) ....3865
69.2.124 (419h) IA32_MC6_STATUS (SA) ....3867
69.2.125 (41Ah) IA32_MC6_ADDR (SA) ....3867
69.2.126 (41Ch) IA32_MC7_CTL (MEM 0)....3868
69.2.127 (41Dh) IA32_MC7_STATUS (MEM 0) ....3870
69.2.128 (41Fh) IA32_MC7_MISC (MEM 0)....3871
69.2.129 (420h) IA32_MC8_CTL (MEM 1)....3872
69.2.130 (421h) IA32_MC8_STATUS (MEM 1)....3874
69.2.131 (423h) IA32_MC8_MISC (MEM 1)....3874
69.2.132 (4C1h) IA32_A_PMC0....3874
69.2.133 (4C2h) IA32_A_PMC1....3875
69.2.134 (4C3h) IA32_A_PMC2....3875
69.2.135 (4C4h) IA32_A_PMC3....3875
69.2.136 (4E0h) MSR_SMM_FEATURE_CONTROL....3875
69.2.137 (4E2h) MSR_SMM_DELAYED....3876
69.2.138 (4E3h) MSR_SMM_BLOCKED ....3876
69.2.139 (500h) IA32_SGX_SVN_STATUS ....3876
69.2.140 (503h) MSR_SGX_DEBUG_MODE ....3877
69.2.141 (560h) IA32_RTIT_OUTPUT_BASE....3877
69.2.142 (561h) IA32_RTIT_OUTPUT_MASK_PTRS ....3877
69.2.143 (570h) IA32_RTIT_CTL....3878
69.2.144 (571h) ARR_CR_RTIT_STATUSIA32_RTIT_STATUS....3878
69.2.145 (572h) IA32_RTIT_CR3_MATCH....3878
69.2.146 (580h) IA32_RTIT_ADDR0_A....3878
69.2.147 (581h) IA32_RTIT_ADDR0_B....3879
69.2.148 (582h) IA32_RTIT_ADDR1_A....3879
69.2.149 (583h) IA32_RTIT_ADDR1_B....3879
69.2.150 (600h) IA32_DS_AREA ....3879
69.2.151 (601h) MSR_VR_CURRENT_CONFIG ....3880
69.2.152 (603h) MSR_VR_MISC_CONFIG ....3880
69.2.153 (606h) MSR_RAPL_POWER_UNIT ....3881
69.2.154 (610h) MSR_PKG_POWER_LIMIT ....3882
69.2.155 (614h) MSR_PKG_POWER_INFO ....3884
69.2.156 (615h) PPL3_CONTROL....3884
69.2.157 (618h) MSR_DRAM_POWER_LIMIT....3885
69.2.158 (61Ch) MSR_DRAM_POWER_INFO....3886
69.2.159 (633h) MSR_PKGC_IRTL3 ....3887
69.2.160 (634h) MSR_PKGC_IRTL4 ....3887
69.2.161 (635h) MSR_PKGC_IRTL5 ....3888
69.2.162 (639h) MSR_PP0_ENERGY_STATUS....3888
69.2.163 (641h) MSR_PP1_ENERGY_STATUS....3889
69.2.164 (64Ch) MSR_TURBO_ACTIVATION_RATIO ....3889
69.2.165 (64Fh) MSR_CORE_PERF_LIMIT_REASONS....3890
69.2.166 (80Fh) IA32_X2APIC_SIVR....3891
69.2.167 (828h) IA32_X2APIC_ESR....3891
69.2.168 (82Fh) IA32_X2APIC_LVT_CMCI ....3892
69.2.169 (830h) IA32_X2APIC_ICR ....3893
69.2.170 (832h) IA32_X2APIC_LVT_TIMER....3894
69.2.171 (833h) IA32_X2APIC_LVT_THERMAL ....3895
69.2.172 (834h) IA32_X2APIC_LVT_PMI ....3895
69.2.173 (835h) IA32_X2APIC_LVT_LINT0 ....3896
69.2.174 (836h) IA32_X2APIC_LVT_LINT1 ....3897
69.2.175 (837h) IA32_X2APIC_LVT_ERROR....3897
69.2.176 (838h) IA32_X2APIC_INIT_COUNT....3898
69.2.177 (839h) IA32_X2APIC_CUR_COUNT....3898

69.2.178 (83Eh) IA32_X2APIC_DIV_CONF ..... 3898
69.2.179 (83Fh) IA32_X2APIC_SELF_IPI ..... 3898
69.2.180 (C80h) IA32_DEBUG_FEATURE ..... 3899
69.2.181 (C8Fh) IA32_PQR_ASSOC ..... 3899
69.2.182 (D10h) IA32_L2_QOS_MASK_0 ..... 3899
69.2.183 (D11h) IA32_L2_QOS_MASK_1 ..... 3900
69.2.184 (D12h) IA32_L2_QOS_MASK_2 ..... 3900
69.2.185 (D13h) IA32_L2_QOS_MASK_3 ..... 3900
69.2.186 (D90h) IA32_BNDCFGS ..... 3901
69.2.187 (DA0h) IA32_XSS ..... 3901
69.2.188 (C0000080h) IA32_EFER ..... 3901
69.2.189 (C0000081h) IA32_STAR ..... 3902
69.2.190 (C0000082h) IA32_LSTAR ..... 3902
69.2.191 (C0000083h) IA32_CSTAR ..... 3902
69.2.192 (C0000084h) IA32_FMASK ..... 3902
69.2.193 (C0000100h) IA32_FS_BASE ..... 3903
69.2.194 (C0000101h) IA32_GS_BASE ..... 3903
69.2.195 (C0000102h) IA32_KERNEL_GS_BASE ..... 3903
69.2.196 (C0000103h) IA32_TSC_AUX ..... 3903

# Figures

1-1 High-Level Block Diagram ..... 113
2-1 What is Covered in This Chapter ..... 131
2-2 Version Information Returned by CPUID in EAX ..... 139
3-1 Host Root Space - PCI Device Map ..... 182
3-2 DWORD Format of CONFIG_ADDRESS at I/O Address CF8h ..... 185
4-1 Strap Loading Flow ..... 192
4-2 Hard Strap Functionality ..... 194
4-3 IE Fuses ..... 260
5-1 What is Covered in This Chapter ..... 261
5-2 Register Map ..... 263
6-1 SoC Machine Check MSRs — Global Control and Bank Registers ..... 271
6-2 SoC Core, Module, and Package Machine Check Registers ..... 273
6-3 Intel Atom® Processor C3000 Product Family Error Reporting ..... 278
6-4 Error Handling Architecture ..... 281
6-5 MCERR and IERR Handling ..... 283
6-6 Local and Global Error Register Structure ..... 284
6-7 Global Error Registers and System Event Reporting ..... 290
6-8 PCIe Error Signaling and Logging ..... 294
6-9 Error Signaling and Logging for PCIe Root Complex Integrated Endpoints ..... 297
6-10 Error Signaling and Logging for Non-PCI Devices ..... 305
6-11 Register Map ..... 320
7-1 What is Covered in This Chapter ..... 325
7-2 SoC Clock Architecture ..... 329
8-1 PIRQ Routing Control Overview ..... 336
8-2 General Purpose Event (GPE) Register Model ..... 340
9-1 What is Covered in This Chapter ..... 347
10-1 What is Covered in This Chapter ..... 359
10-2 FIA and HSIO Structure ..... 362
10-3 Intel® ME-BIOS/UEFI Interactions in a System with Intel® SPS Firmware ..... 364
10-4 PCIe x1 Lane and SATA Example ..... 366
10-5 Register Map ..... 367
11-1 What is Covered in This Chapter ..... 369
11-2 PCIe Bifurcation and Physical Ports ..... 380
11-3 Register Map ..... 395
12-1 What is Covered in This Chapter ..... 396
12-2 Register Map ..... 401
13-1 What is Covered in This Chapter ..... 410
13-2 System Architecture and Interface ..... 421
13-3 Error Reporting Mechanism ..... 439
13-4 Shared SPI Flash Shadow RAM ..... 447
13-5 10GBASE-T Implementation ..... 465
13-6 10G Backplane Implementation Using KR to Backplane (Switch) ..... 465
13-7 SFP+ Implementation ..... 466
13-8 1G BASE-T Implementation ..... 467
13-9 Basic PHY/MAC Connectivity ..... 468
13-10 MDIO Timing Source by the MAC ..... 469
13-11 MDIO Timing Sourced by the PHY ..... 469
13-12 Behavior of MDIO During TA Field of a Read Transaction ..... 472
13-13 802.3X Link Flow Control (PAUSE) ..... 481
13-14 Register Map ..... 492
14-1 What is Covered in This Chapter ..... 493
14-2 USB 3.0 Cable ..... 496
14-3 Universal Serial Bus, Revision 3.0 System Block Diagram ..... 498

14-4 USB 3.0 eXtensible Host Controller ..... 500
14-5 General Architecture of the eXtensible Host Controller Interface ..... 501
14-6 Register Map ..... 508
15-1 What is Covered in This Chapter ..... 522
16-1 What is Covered in This Chapter ..... 525
16-2 Register Map ..... 553
17-1 What is Covered in This Chapter ..... 557
17-2 Register Map ..... 560
18-1 What is Covered in This Chapter ..... 565
18-2 UART MUXed Signals ..... 567
18-3 UART Core Block Diagram Overview ..... 569
18-4 UART Data Transfer Flow ..... 572
18-5 7-bit Character Encoding ..... 572
18-6 DMA Block Diagram ..... 578
18-7 Logical Representation of Channel Arbitration ..... 579
18-8 Logical Representation of Descriptor Arbitration ..... 580
18-9 Register Map ..... 587
19-1 What is Covered in This Chapter ..... 592
19-2 Connection to the SPI Devices ..... 594
19-3 Flash Regions ..... 599
19-4 Organize Flash Region Single Flash vs. Dual-Flash Components ..... 599
19-5 Flash Descriptor Sections ..... 600
19-6 Dual-Output-Fast-Read Timing ..... 613
19-7 Dual-I/O-Fast-Read Timing ..... 614
19-8 Quad Output Fast Read Timing ..... 615
19-9 Quad-I/O-Fast-Read Timing ..... 616
19-10 Register Map ..... 622
20-1 What is Covered in This Chapter ..... 626
20-2 LPC_CLKRUN_N Timings ..... 630
20-3 Register Map ..... 631
21-1 What is Covered in This Chapter ..... 638
21-2 Master and Slave PIC ..... 639
21-3 MSI-Like IOxAPIC Interrupt Message for the Local APICs ..... 647
21-4 Register Map ..... 650
22-1 What is Covered in This Chapter ..... 654
22-2 Register Map ..... 656
23-1 What is Covered in This Chapter ..... 659
23-2 Programmable Interval Timer (PIT) Implementation ..... 660
23-3 Register Map ..... 665
24-1 What is Covered in This Chapter ..... 667
24-2 Register Map ..... 673
25-1 What is Covered in This Chapter ..... 675
25-2 HPET Counter and Timers ..... 676
25-3 Register Map ..... 682
26-1 What is Covered in This Chapter ..... 685
26-2 Custom GPIO Circuit Definition ..... 691
26-3 GPIO Logic Overview Diagram ..... 695
27-1 Arrangement of SMBus in SoC ..... 708
28-1 What is Covered in This Chapter ..... 711
28-2 Register Map ..... 725
29-1 What is Covered in This Chapter ..... 729
29-2 ARP-Capable (Slave) Device Behavior Flow Diagram ..... 736
29-3 ARP Master Behavior Flow Diagram ..... 743
29-4 Master Descriptor Ring Buffer ..... 750
29-5 Master Descriptor Format ..... 750

29-6 Hardware-Firmware Flow Diagram—DMA Mode ..... 760
29-7 Target Ring Buffer ..... 766
29-8 Target Header Format ..... 766
29-9 High-Level Target Flow ..... 775
29-10 Host Notify Target Flow ..... 777
29-11 SMBus ARP Target Flow ..... 778
29-12 General Purpose Block Read with PEC Target Flow ..... 780
29-13 SMBus/$I^2C$ Target Flow ..... 782
29-14 Target Dynamic Policy Update ..... 784
29-15 MCTP Over SMBus Packet Format ..... 792
29-16 Register Map ..... 794
30-1 What is Covered in This Chapter ..... 795
30-2 SMBus Protocol ..... 797
30-3 SMBus Block Write Command ..... 798
30-4 SMBus Block Read Command ..... 798
30-5 PECI Message Header in the SMBus Packet ..... 799
30-6 PECI Write-Read Protocol ..... 800
30-7 PECI Device Info Field Definition ..... 811
30-8 PCI Configuration Address Format ..... 821
30-9 Channel Index and DIMM Index Parameter Word ..... 835
30-10 Write DRAM Rank Temperature Data Dword ..... 836
30-11 Read DIMM Temperature Data Dword ..... 836
30-12 Read DRAM Channel Temperature Data Dword ..... 837
30-13 Read DRAM Accumulated Energy Data Dword ..... 837
30-14 Read DRAM Power Info Data Dword ..... 838
30-15 DRAM Power Limit Data ..... 839
30-16 DRAM Power Limit Performance Data ..... 840
30-17 CPU ID Data ..... 848
30-18 Platform ID Data ..... 848
30-19 PCU Device ID Data ..... 849
30-20 Maximum Thread ID Data ..... 849
30-21 Processor Microcode Revision ..... 849
30-22 Package Power SKU Unit Data ..... 850
30-23 Package Power SKU Data ..... 851
30-24 Package Temperature Read Data ..... 852
30-25 Temperature Target Read ..... 853
30-26 Package Thermal Status Read/Clear ..... 853
30-27 Current Limit Read Data ..... 854
30-28 Accumulated Energy Read Data ..... 854
30-29 Package Turbo Power Limit Data ..... 855
30-30 Package Power Limit Performance Data ..... 856
30-31 Efficient Performance Indicator Data ..... 856
30-32 ACPI P-T Notify Data ..... 857
30-33 PL2 and PL3 ..... 858
30-34 PL3 Register Definition ..... 858
30-35 Read Ratio Constraints Parameter ..... 859
30-36 Read Ratio Constraints Return Value ..... 859
30-37 Read Ratio Limit ..... 860
30-38 Write Ratio Limit ..... 860
30-39 Crash Dump Capabilities Details ..... 861
30-40 Crash Dump Capabilities (Parameter0 = 0x1) ..... 861
30-41 Request Parameters ..... 862
30-42 Response Data for Requested Frame ..... 862
30-43 PECI Device Temp [15:0] Format - Temperature Sensor Data ..... 871
31-1 What is Covered in This Chapter ..... 873

31-2 Intel® QAT Usage Block Diagram ..... 875
31-3 Direct I/O Assignment Model with IOV Support ..... 877
31-4 EP Functional Description Block Diagram ..... 879
31-5 Register Map ..... 890
33-1 ACPI Cold Boot Sequence ..... 898
33-2 Non-ACPI Cold Boot Sequence ..... 899
33-3 ACPI Shutdown Sequence ..... 900
33-4 Non-ACPI Shutdown Sequence ..... 901
34-1 What is Covered in This Chapter ..... 903
34-2 Global System Power States and Transitions ..... 911
34-3 Processor Power States ..... 913
35-1 TM1 Use of Processor Clock Duty Cycle Modulation ..... 927
35-2 CLTT SMBus Connection to DIMMs ..... 937
36-1 What is Covered in This Chapter ..... 947
36-2 Register Map ..... 950
37-1 What is Covered in This Chapter ..... 964
37-2 Intel Atom® Processor C3000 Product Family Embedded Sub-System ..... 966
37-3 Sample IE Usages ..... 967
37-4 Register Map ..... 972
38-1 What is Covered in This Chapter ..... 985
39-1 Ball Quadrant Map ..... 992
40-1 SPI Timing Diagram ..... 1053
40-2 Dual-Output Fast Read Timing with 24-bit Addressing ..... 1054
40-3 Dual I/O Fast Read Timing with 24-bit Addressing ..... 1055
40-4 Quad Output Fast Read Timing with 24-bit Addressing ..... 1056
40-5 Quad I/O Fast Read Timing with 24-bit Addressing ..... 1057
40-6 1.05V GPIO Receiver Timing ..... 1062
40-7 1.05V GPIO Driver Timing ..... 1063
40-8 1.8V GPIO Receiver Timing ..... 1064
40-9 3.3V GPIO Driver Timing ..... 1067
42-1 SoC Package Drawing ..... 1076

# Tables

1-1 Features by product SKU - Server and Cloud Storage SKUs 0 through 4 ..... 107
1-2 Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10 ..... 109
1-3 Features by Product SKU - Extended Temperature SKUs 11 through 14 ..... 111
1-4 Terminology ..... 121
1-5 Reference Documents ..... 127
2-1 CPUID Leaf 0h ..... 140
2-2 CPUID Leaf 1h, Output EAX and EBX ..... 141
2-3 CPUID Leaf 1h, Output ECX ..... 142
2-4 CPUID Leaf 1h, Output EDX ..... 144
2-5 CPUID Leaf 2h ..... 146
2-6 CPUID Leaf 4h ..... 147
2-7 CPUID Leaf 5h ..... 149
2-8 CPUID Leaf 6h ..... 150
2-9 CPUID Leaf 7h ..... 151
2-10 CPUID Leaf Ah ..... 153
2-11 CPUID Leaf Bh ..... 154
2-12 CPUID Leaf Dh ..... 155
2-13 CPUID Leaf 10h ..... 157
2-14 CPUID Leaf 14h ..... 159
2-15 Basic CPUID Leaf 15h ..... 160
2-16 Extended CPUID Leaf 80000000h ..... 161
2-17 Extended CPUID Leaf 80000001h ..... 162
2-18 Extended CPUID Leaves 80000002h, 80000003h, and 80000004h ..... 163
2-19 Extended CPUID Leaf 80000006h ..... 163
2-20 Extended CPUID Leaf 80000007h ..... 164
2-21 Extended CPUID Leaf 80000008h ..... 164
3-1 Host Memory Space Address Map ..... 166
3-2 Relocatable MMIO in Host Memory Space ..... 169
3-3 Host Fixed I/O Address Space ..... 176
3-4 Host Relocatable I/O Address Space ..... 180
3-5 Host PCI Bus, Device, Function Numbers and Device IDs ..... 183
3-6 Configuration Registers Used to Configure Host Memory Space ..... 188
3-7 Relocatable Host Memory-Mapped Aperture to Access Certain System Agent Registers ..... 190
3-8 Fixed-Address Host I/O-Mapped Registers to Access Configuration Space ..... 190
4-1 SoC Hard Straps/Pin Straps and Descriptions ..... 195
4-2 Boot interface Hard Strap/ Pin Strap Configurations ..... 198
4-3 Flash Descriptor Soft Straps ..... 201
5-1 References ..... 261
5-2 Registers in Host Configuration Space ..... 264
6-1 Signal Names and Descriptions ..... 265
6-2 References ..... 266
6-3 SoC Machine Check MSR Addresses ..... 272
6-4 MSR_SMM_MCA_CONTROL (MSR 52h) — Enhanced MCA Control ..... 274
6-5 Local Error Register General Descriptions ..... 285
6-6 List of SoC Global Error Registers ..... 286
6-7 Local Error Registers — PCI Express Root Ports ..... 299
6-8 Local Error Registers — PCI Express RCEC and RCiEP Functions ..... 300
6-9 Local Error Registers — PCI Express Functions Attached to Virtual Root Ports ..... 301
6-10 Local Error Registers — SATA, USB, LPC and SMBus-Legacy ..... 303
6-11 Local Error Registers — SPI and HSUART ..... 303
6-12 Local Error Registers — A-Unit and B-Unit ..... 306
6-13 Local Error Registers — D-Units ..... 307

6-14 Local Error Registers — Integrated Error Handler (IEH) ..... 309
6-15 Summary of Default Error Logging and Responses ..... 310
6-16 Registers in Host Configuration Space ..... 321
6-17 Registers in Host Configuration Space ..... 323
7-1 Signal Names and Descriptions ..... 326
8-1 PIRQ Routing Control Information For INT[A-D] Messages ..... 333
8-2 PIRQA through PIRQH Routing Register IRQ Decode ..... 337
8-3 Routing of SCI to the I/O APIC ..... 339
8-4 GPE0_DW Group Mapping ..... 341
8-5 Access to the ACPI General Purpose Event Registers ..... 342
8-6 I/O APIC Input Mapping ..... 343
8-7 8259 PIC Input Mapping ..... 345
8-8 Device Interrupt / Event Sources ..... 346
9-1 References ..... 347
9-2 Signal Names and Descriptions ..... 348
9-3 DIMM Configurations, Two DIMM Socket per Channel ..... 357
10-1 Signal Names and Descriptions ..... 360
10-2 Bifurcation Control for PCIe RP Clusters 0 and 1 ..... 365
10-3 Configuration and Capabilities Register Map ..... 368
10-4 Sideband Registers ..... 368
11-1 References ..... 370
11-2 Signal Names and Descriptions ..... 371
11-3 PCI Express Features ..... 372
11-4 PCI Express Transaction ID Handling ..... 382
11-5 PCI Express Attribute Handling ..... 383
11-6 PCI Express CompleterID Handling ..... 384
11-7 Transaction Ordering on PCIe Block ..... 385
11-8 Bridge Mapping for Interrupt Virtual Wires ..... 387
12-1 References ..... 396
12-2 Signal Names and Descriptions ..... 397
12-3 SATA Controller Operations Summary ..... 399
12-4 Interaction Between INTx# and MSI ..... 400
12-5 Configuration and Capabilities Registers ..... 402
12-6 Memory Mapped Registers ..... 404
12-7 I/O Mapped Registers ..... 409
13-1 References ..... 411
13-2 Signal Names and Descriptions ..... 412
13-3 Supported System Configurations ..... 416
13-4 Network Features ..... 417
13-5 Host Interface Features ..... 417
13-6 Miscellaneous Features ..... 418
13-7 LAN Functions Features ..... 418
13-8 LAN Performance Features ..... 419
13-9 Virtualization Features ..... 420
13-10 Manageability Features ..... 420
13-11 PCI and PCIe Capabilities Supported ..... 422
13-12 Base Address Registers ..... 423
13-13 Tx Data Flow ..... 424
13-14 Rx Data Flow ..... 425
13-15 Link Modes ..... 427
13-16 Transaction Types Accepted by the Transaction Layer ..... 432
13-17 Transaction Types Initiated by the Transaction Layer ..... 434
13-18 Supported Message in the LAN Controllers (as a Receiver) ..... 436
13-19 Response and Reporting of PCIe Error Events ..... 440
13-20 PCIe Errors Reported to Device Software ..... 443

13-21 Clients and Access Types to the Shared SPI Flash ..... 449
13-22 Small Resource Structure ..... 453
13-23 Large Resource Structure ..... 453
13-24 VPD Structure ..... 454
13-25 SPI Flash Module Encoding ..... 459
13-26 GPI to SDP Bit Mappings ..... 461
13-27 SDP Settings ..... 461
13-28 LED Control Source (MODE) Mapping ..... 463
13-29 MDC Frequency as Function of Link Speed and MDC Speed Bit ..... 469
13-30 Clause 22 Basic MDIO Frame Format ..... 470
13-31 Clause 45 Indirect Addressing MDIO Frame Format ..... 471
13-32 Direct $I^2C$ Interface Control ..... 475
13-33 $I^2C$ Read Transaction - Dummy Write ..... 475
13-34 $I^2C$ Read Transaction - Word Read ..... 475
13-35 $I^2C$ Read Transaction - Dummy Write ..... 475
13-36 $I^2C$ Read Transaction - Byte Read ..... 475
13-37 $I^2C$ Write Transaction - Word Write ..... 476
13-38 $I^2C$ Write Transaction - Byte Write ..... 476
13-39 PHY Access Registers ..... 477
13-40 PHY Registers' Addressing ..... 477
13-41 MAC Control Frame Format ..... 480
13-42 802.3X Packet Format ..... 480
13-43 Transfer of PAUSE Packet to Host (DPF Bit) ..... 484
13-44 LAN Controllers Delay Values (DV) Used For FCRTH ..... 487
13-45 LAN Controllers FCRTL ..... 488
13-46 Some Recommended Rx Packet Buffer Settings ..... 489
13-47 Pacing Speeds at 10 GbE Link Speed ..... 490
14-1 References ..... 493
14-2 Signal Names and Descriptions ..... 494
14-3 Transaction Type Supported by USB Block ..... 505
14-4 Reset Signal Information ..... 505
14-5 USB 2.0 Link States and Power Actions ..... 506
14-6 USB 3.0 Link States and Power Actions ..... 507
14-7 Configuration and Capabilities Register Map ..... 509
14-8 Relocatable Memory-Mapped Registers ..... 511
15-1 Platform Control Cluster (PCC) ..... 523
16-1 Signal Names and Descriptions ..... 526
16-2 SoC Circuitry Powered-On During the G2 (S5) Global Power State ..... 529
16-3 SoC Circuitry Powered-On During the G3 Global Power State ..... 529
16-4 Causes of SMI and SCI ..... 533
16-5 Power Management Wake Events ..... 537
16-6 Power Management Event Enables ..... 541
16-7 Causes of Host Reset and Global Reset ..... 544
16-8 Configuration and Capabilities Register Map ..... 554
16-9 Relocatable Memory-Mapped Registers ..... 554
16-10 Relocatable I/O-Mapped Registers - Power Management Controller (PMC) ..... 555
16-11 Fixed-Address I/O-Mapped Registers ..... 555
16-12 Relocatable I/O-Mapped Registers - Legacy SMBus Controller ..... 556
16-13 Sideband Registers - Power Management Control (PMC) ..... 556
16-14 Sideband Registers - Legacy SMBus (for TCO) ..... 556
17-1 References ..... 557
17-2 Signal Names and Descriptions ..... 558
17-3 Configuration and Capabilities Register Map ..... 561
17-4 Memory-Mapped Registers ..... 562
18-1 References ..... 565

18-2 Signal Names and Descriptions ..... 566
18-3 Legacy COM I/O Ports ..... 570
18-4 Supported Baud Rates ..... 576
18-5 Interrupt Enabling ..... 577
18-6 UART Memory Space ..... 577
18-7 Interrupt Enabling ..... 582
18-8 PCI Configuration Registers (HSUART_REGS_PCI) ..... 588
18-9 UART and DMA Memory Space ..... 589
18-10 UART Memory Mapped Registers ..... 589
18-11 DMA Channel Memory Mapped Registers ..... 590
18-12 UART I/O Mapped Registers ..... 591
19-1 References ..... 592
19-2 Signal Names and Descriptions ..... 593
19-3 Flash Regions ..... 598
19-4 Flash Protection Mechanism Summary ..... 605
19-5 Programmed Access Error Status ..... 607
19-6 Hardware Sequencing Opcodes ..... 609
19-7 Address and Data Transfer for TPM ..... 619
19-8 Configuration and Capabilities Register Map ..... 623
19-9 BIOS Flash Program Register Address Map ..... 624
20-1 References ..... 626
20-2 Signal Names and Descriptions ..... 627
20-3 LPC_CLKRUN_N Timings ..... 630
20-4 LPC Register Map - Configuration and Capabilities Register Map ..... 632
20-5 LPC Claimable Fixed I/O Range as Target on Primary Interface ..... 633
20-6 LPC Claimable Configurable I/O Range as Target on Primary Interface ..... 635
20-7 LPC Claimable Memory Mapped Range as Target on Primary Interface ..... 636
20-8 LPC Claimable Configurable I/O Range as Target on Primary Interface ..... 637
20-9 BIOS Boot Decode Claiming Agent ..... 637
21-1 Interrupt Status Registers ..... 639
21-2 Content of Interrupt Vector Byte ..... 640
21-3 IOxAPIC Interrupt Message Address ..... 648
21-4 IOxAPIC Interrupt Message Data ..... 648
21-5 Configuration and Capabilities Register Map ..... 651
21-6 IOxAPIC Register Access Window in Host Memory Space ..... 652
21-7 IOxAPIC Registers ..... 652
21-8 IOxAPIC Interrupt Messages for Local APICs in Host Memory Space ..... 652
21-9 8259 (PIC) Registers in Host I/O Space ..... 653
22-1 Configuration and Capabilities Register Map ..... 657
22-2 Relocatable Memory-Mapped Registers ..... 658
23-1 Counter Operating Modes ..... 662
23-2 Fixed-Address I/O-Mapped Registers ..... 666
24-1 Signal Names and Descriptions ..... 668
24-2 RTC Standard and Extended Static RAM Banks ..... 670
24-3 Relocatable Memory-Mapped Registers ..... 674
24-4 Fixed-Address I/O-Mapped Registers ..... 674
25-1 References ..... 675
25-2 Legacy Replacement Routing ..... 678
25-3 Fixed-Address Memory-Mapped Registers ..... 683
26-1 GPIO Ball and Pad Name ..... 686
26-2 Pin Name Descriptions ..... 693
26-3 Input Signal Configuration ..... 695
26-4 Output Signal Configuration ..... 695
26-5 GPIO Pad Configuration Description ..... 698
26-6 PMC pci_cfg.PWRMBASE ..... 704

26-7 PMC pci_cfg.ABASE ..... 705
27-1 References ..... 706
28-1 References ..... 711
28-2 Signal Names ..... 712
28-3 Supported SMBus ARP, SMBus, and I2C Protocols ..... 715
28-4 I2C Block Read ..... 721
28-5 Enable for SMBALERT# (SMB_LEG_ALRT_N) ..... 723
28-6 Enables for Legacy SMBus Controller Events ..... 723
28-7 Enables for a Received Notify ARP Master Command ..... 723
28-8 Registers in the Configuration Space ..... 726
28-9 Registers in the Memory Space ..... 727
28-10 Registers in the I/O Space ..... 728
29-1 References ..... 729
29-2 Signal Names ..... 730
29-3 List of Supported SMBus ARP, SMBus, and I2C Protocols ..... 732
29-4 ARP Nomenclature ..... 734
29-5 Device Decodes of AV and AR Flags ..... 734
29-6 UDID Format ..... 735
29-7 ARP Slave Operations ..... 737
29-8 Hardware Decoding of ARP, SMBus, and I2C Target Transactions ..... 740
29-9 Hardware/Firmware Response to SMBus and ARP Protocols ..... 741
29-10 ARP Master Operation ..... 744
29-11 ARP Initialization Flow ..... 746
29-12 SMT Timing Mode Maximum Clock Frequency Ranges ..... 748
29-13 Master Descriptor Field Descriptions ..... 751
29-14 SMBus Transaction Encodings ..... 754
29-15 I2C Commands ..... 757
29-16 DIMM SPD EEPROM Write-Disable Mechanism ..... 764
29-17 Target Header Descriptor ..... 767
29-18 Valid Target Descriptor MTYPE and TTYPE Combinations ..... 769
29-19 Target Header Encodings (TSTS) Per Transaction Type (TTYPE) ..... 770
29-20 Target Transaction Behavior Due to SUSCHKB.IRWST ..... 776
29-21 Summary of SMT Interrupt Enables and Sources ..... 785
29-22 Error MSI Scheduling ..... 788
29-23 Interrupt Cause Information ..... 789
29-24 SMT Soft Reset Exceptions ..... 790
29-25 SMT Function Level Reset Exceptions ..... 791
29-26 MCTP Over SMBus Packet Format ..... 793
30-1 References ..... 795
30-2 Signal Names ..... 796
30-3 SMBus Write Commands ..... 798
30-4 SMBus Read Command ..... 798
30-5 PECI Proxy Command Protocol Format ..... 801
30-6 PECI Proxy Read ..... 802
30-7 Supported PECI Commands ..... 806
30-8 Ping - PECI Proxy Block Write ..... 807
30-9 Ping - PECI Proxy Block Read ..... 808
30-10 GetDIB() PECI Proxy Block Write ..... 809
30-11 GetDIB() PECI Proxy Block Read ..... 810
30-12 PECI Revision Number Definition ..... 811
30-13 GetTemp() PECI Proxy Block Write ..... 812
30-14 GetTemp() PECI Proxy Block Read ..... 813
30-15 RdPkgConfig() PECI Proxy Block Write ..... 814
30-16 RdPkgConfig() PECI Proxy Block Read ..... 815
30-17 WrPkgConfig() PECI Proxy Block Write ..... 816

30-18 WrPkgConfig() PECI Proxy Block Read ..... 817
30-19 RdIAMSR() PECI Proxy Block Write ..... 819
30-20 RdIAMSR() PECI Proxy Block Read ..... 820
30-21 RdPCIConfigLocal() PECI Proxy Block Write ..... 822
30-22 RdPCIConfigLocal() PECI Proxy Block Read ..... 823
30-23 WrPCILocalConfig() PECI Proxy Block Write ..... 824
30-24 WrPCILocalConfig() PECI Proxy Block Read ..... 826
30-25 RdEndPointConfig() PECI Proxy Block Write ..... 827
30-26 RdEndPointConfig() PECI Proxy Block Read ..... 828
30-27 WrEndPointConfig() PECI Proxy Block Write ..... 829
30-28 WrEndPointConfig() PECI Proxy Block Read ..... 830
30-29 MCA Access via RdEndPointConfig() ..... 831
30-30 Endpoints Supporting Rd/WrPCIConfigLocal ..... 832
30-31 Endpoints Supporting Rd/WrEndPointConfig ..... 833
30-32 Summary of DRAM Thermal and Power Optimization Services ..... 834
30-33 Channel Index and DIMM Index ..... 835
30-34 Summary of CPU Thermal and Power Optimization Services ..... 841
30-35 Power Control Register Unit Calculations ..... 850
30-36 Response Data for Requested Items ..... 863
30-37 RdPkgConfig() PECI Proxy Block Write ..... 866
30-38 RdPkgConfig() PECI Proxy Block Read ..... 867
30-39 Core Crash Details ..... 869
30-40 Error Codes ..... 872
31-1 Ring Assignment to Intel® QAT Virtual Functions ..... 878
31-2 PCIe Commands Supported by RI ..... 880
31-3 PCIe EP Function Mapping ..... 880
31-4 EP Mapping of BARs to MMIOs ..... 881
31-5 Bus, Device and Function Interpretation ..... 881
31-6 PCIe Error Sources ..... 884
31-7 Device-Specific Errors ..... 889
32-1 Signal Descriptions ..... 893
33-1 References ..... 895
33-2 Signal Names and Descriptions ..... 895
33-3 SoC Required Power Supplies ..... 896
33-4 SoC Power Supply for each Supported Power State ..... 896
33-5 Supported ACPI Transitions And Definitions ..... 897
34-1 sVID Controller Signals ..... 903
34-2 References ..... 904
34-3 SoC Voltage Rails ..... 907
34-4 sVID Commands ..... 909
34-5 ACPI Power States ..... 911
34-6 ACPI Power State Transitions for the SoC ..... 912
34-7 Core C-States ..... 914
34-8 Package C-States ..... 914
34-9 ACPI P-State Mappings ..... 915
34-10 I/O Power Management Summary ..... 918
35-1 Signal Names and Descriptions ..... 919
35-2 MSR_POWER_CTL (MSR 1FCh) ..... 924
35-3 PROCHOT_N Signal Direction and Function ..... 925
35-4 MSR Registers ..... 946
36-1 Signal Names and Descriptions ..... 948
36-2 Configuration and Capabilities Register Map ..... 951
36-3 Relocatable Memory-Mapped Registers ..... 953
36-4 Configuration and Capabilities Register Map ..... 954
36-5 Relocatable Memory-Mapped Registers ..... 955

36-6 Configuration and Capabilities Register Map ..... 956
36-7 Relocatable I/O-Mapped Registers ..... 957
36-8 Configuration and Capabilities Register Map ..... 959
36-9 Relocatable Memory-Mapped Registers ..... 960
36-10 Configuration and Capabilities Register Map ..... 961
36-11 Relocatable Memory-Mapped Registers ..... 962
36-12 Relocatable I/O-Mapped Registers ..... 963
37-1 Signal Names and Descriptions ..... 965
37-2 Configuration and Capabilities Register Map ..... 973
37-3 Relocatable Memory-Mapped Registers ..... 975
37-4 Configuration and Capabilities Register Map ..... 976
37-5 Relocatable Memory-Mapped Registers ..... 977
37-6 Configuration and Capabilities Register Map ..... 978
37-7 Relocatable I/O-Mapped Registers ..... 979
37-8 Configuration and Capabilities Register Map ..... 981
37-9 Relocatable Memory-Mapped Registers ..... 982
37-10 Configuration and Capabilities Register Map ..... 983
37-11 Relocatable Memory-Mapped Registers ..... 984
37-12 Relocatable I/O-Mapped Registers ..... 984
38-1 References ..... 985
38-2 Signal Names and Descriptions ..... 986
39-1 Signal Pin Name Directory - Sorted by Name ..... 995
39-2 Treatment of Reserved Pins ..... 1026
39-3 Signals that Share the Twenty HSIO Lanes ..... 1028
39-4 Voltage Pins Sorted by Platform VR and SoC Voltage Group ..... 1029
39-5 Package Rotation (ROT) Pins ..... 1032
39-6 Directory of Signal and Voltage Pins - Sorted by Pin Number ..... 1033
40-1 SPI (17.14MHz) Signal Timing Specifications ..... 1052
40-2 SPI (30MHz) Signal Timing Specifications ..... 1052
40-3 SPI (48MHz) Signal Timing Specifications ..... 1053
40-4 CS# Setup and CS# Hold ..... 1054
40-5 1.05V GPIO Receiver DC Characteristics ..... 1062
40-6 1.05V GPIO Receiver AC Characteristics ..... 1062
40-7 1.05V GPIO Driver DC Characteristics ..... 1063
40-8 1.05V GPIO Driver AC Characteristics ..... 1063
40-9 1.8V GPIO Receiver DC Characteristics ..... 1064
40-10 1.8V GPIO Receiver AC Characteristics ..... 1064
40-11 1.8V GPIO Driver DC Characteristics ..... 1065
40-12 1.8V GPIO Driver AC Characteristics ..... 1065
40-13 1.8V GPIO Driver Timing ..... 1065
40-14 3.3V GPIO Receiver DC Characteristics ..... 1066
40-15 3.3V GPIO Receiver AC Characteristics ..... 1066
40-16 3.3V GPIO Receiver Timing ..... 1066
40-17 3.3V GPIO Driver DC Characteristics ..... 1067
40-18 3.3V GPIO Driver AC Characteristics ..... 1067
41-1 Storage Condition Ratings - Prior to Board-Attach ..... 1069
41-2 Component Temperature Range ..... 1070
41-3 Voltage Supply Requirements Under Normal Operating Conditions ..... 1072
41-4 SoC VCC Load Line ..... 1073
41-5 Supply Current Required - Server and Cloud Storage SKUs 0, 1, and 2 (See Table 1-1) ..... 1074
41-6 Supply Current Required - Network & Enterprise Storage SKUs 3, 4, 5, 6, and 7 (See Table 1-2) and eTEMP SKUs 11 and 12 (See Table 1-3) ..... 1074

41-7 Supply Current Required - Network & Enterprise Storage SKUs 8 and 9 (See Table 1-2) ..... 1075
41-8 Supply Current Required - Network & Enterprise Storage SKU 10 (See Table 1-2) and eTEMP SKU 13 and 14 (See Table 1-3) ..... 1075
43-1 Register Attributes Definitions ..... 1081
44-1 Summary of PCI Configuration Registers—0/0/0 ..... 1084
44-2 Summary of Memory Mapped I/O Registers for C-Unit—MCHBAR ..... 1085
44-3 Summary of Memory Mapped I/O Registers for A-Unit—MCHBAR ..... 1086
44-4 Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR ..... 1087
44-5 Summary of Memory Mapped I/O Registers for P-Unit—MCHBAR ..... 1093
44-6 Summary of Memory Mapped I/O Registers—DEFVTDBAR ..... 1096
44-7 Summary of Memory Mapped I/O Registers—DEFVTDBAR ..... 1097
44-8 Summary of Sideband Registers—0x10 (Memory Controller 0) ..... 1099
44-9 Summary of Sideband Registers—0x12 (Memory Controller 1) ..... 1101
44-10 Summary of Sideband Registers—0x99 ..... 1102
45-1 Summary of PCI Configuration Registers—0/4/0 ..... 1482
46-1 Summary of PCI Configuration Registers—0/5/0 ..... 1508
47-1 Summary of PCI Configuration Registers—0/6/0, 0/22/0, and 0/23/0 ..... 1531
48-1 Summary of PCI Configuration Registers—0/9-12, 14-17/0 ..... 1592
48-2 Summary of Sideband Registers—0xB4 (Devices 9-12) ..... 1599
48-3 Summary of Sideband Registers—0xB3 (Devices 14-17) ..... 1599
49-1 Summary of PCI Configuration Registers—0/18/0 ..... 1818
49-2 Summary of Memory Mapped I/O Registers—SMTBAR ..... 1819
50-1 Summary of PCI Configuration Registers—0/19 (or 20)/0 ..... 1871
50-2 Summary of Memory Mapped I/O Registers—MXTBA ..... 1873
50-3 Summary of Memory Mapped I/O Registers—ABAR ..... 1874
51-1 Summary of PCI Configuration Registers—0/21/0 ..... 1996
51-2 Summary of Memory Mapped I/O Registers—MBAR ..... 1998
52-1 Summary of PCI Configuration Registers—0/24/0 ..... 2222
52-2 Summary of PCI Configuration Registers—0/24/1 ..... 2224
52-3 Summary of PCI Configuration Registers—0/24/4 ..... 2225
52-4 Summary of PCI Configuration Registers—0/24/3 ..... 2226
52-5 Summary of PCI Configuration Registers—0/24/2 ..... 2227
52-6 Summary of Memory Mapped I/O Registers—HECI1_MMIO_MBAR ..... 2228
52-7 Summary of Memory Mapped I/O Registers—HECI2_MMIO_MBAR ..... 2229
52-8 Summary of Memory Mapped I/O Registers—HECI3_MMIO_MBAR ..... 2230
52-9 Summary of Memory Mapped I/O Registers—KT_HOST_MEMBAR ..... 2231
52-10 Summary of Memory Mapped I/O Registers—Fixed Addresses ..... 2232
52-11 Summary of I/O Registers—KT_HOST_IOBAR ..... 2233
52-12 Summary of I/O Registers—IDE_HOST_PCMDIOBAR ..... 2234
52-13 Summary of I/O Registers—IDE_HOST_PCTLIOBAR ..... 2235
52-14 Summary of I/O Registers—IDE_HOST_BMIOBAR ..... 2236
52-15 Summary of Sideband Registers for KT (Host)—0xe4 ..... 2237
52-16 Summary of Sideband Registers for IDE (Host)—0xe4 ..... 2238
53-1 Summary of PCI Configuration Registers—0/26/0,1,2 ..... 2383
53-2 Summary of Memory Mapped I/O Registers—MEMBA ..... 2384
53-3 Summary of I/O Registers—IOBA ..... 2386
54-1 Summary of PCI Configuration Registers HECI1—0/27/0 ..... 2438
54-2 Summary of PCI Configuration Registers HECI2—0/27/1 ..... 2440
54-3 Summary of PCI Configuration Registers HECI3—0/27/4 ..... 2441
54-4 Summary of PCI Configuration Registers KT (Host)—0/27/3 ..... 2442
54-5 Summary of PCI Configuration Registers—0/27/2 ..... 2443
54-6 Summary of Memory Mapped I/O Registers—HECI1_MMIO_MBAR ..... 2444
54-7 Summary of Memory Mapped I/O Registers—HECI2_MMIO_MBAR ..... 2445
54-8 Summary of Memory Mapped I/O Registers—HECI3_MMIO_MBAR ..... 2446

54-9 Summary of Memory Mapped I/O Registers—KT_HOST_MEMBAR ....2447
54-10 Summary of Memory Mapped I/O Registers—Fix Addresses ....2448
54-11 Summary of I/O Registers—KT_HOST_IOBAR ....2449
54-12 Summary of I/O Registers—IDE_HOST_PCMDIOBAR....2450
54-13 Summary of I/O Registers—IDE_HOST_PCTLIOBAR....2451
54-14 Summary of I/O Registers—IDE_HOST_BMIOBAR....2452
54-15 Summary of Sideband Registers for KT (Host)—0xf4....2453
54-16 Summary of Sideband Registers IDE (Host)—0xf4....2453
55-1 Summary of Memory Mapped I/O Registers—BAR or BAR1 ....2596
56-1 Summary of PCI Configuration Registers—0/31/0....2644
56-2 Summary of Sideband Registers—0xD2....2644
57-1 Summary of PCI Configuration Registers—0/31/1....2661
58-1 Summary of PCI Configuration Registers—0/31/2....2684
58-2 Summary of Memory Mapped I/O Registers—PWRMBASE ....2685
58-3 Summary of I/O Registers—ABASE....2686
58-4 Summary of I/O Registers—Fixed Addresses....2687
58-5 Summary of Sideband Registers—0x82....2688
59-1 Summary of PCI Configuration Registers—0/31/4....2749
59-2 Summary of Memory Mapped I/O Registers—SMBMBAR ....2750
59-3 Summary of I/O Registers—SBA ....2751
59-4 Summary of I/O Registers—TCOBASE....2752
59-5 Summary of Sideband Registers—0xCF....2753
59-6 Summary of Sideband Registers—0xCC....2754
60-1 Summary of PCI Configuration Registers—0/31/5....2812
60-2 Summary of Memory Mapped I/O Registers—BIOS_SPI_BAR0....2813
60-3 Summary of Sideband Registers—0x93....2814
61-1 Summary of I/O Registers—Fixed I/O Ports ....2875
62-1 Summary of I/O Registers—Fixed Address....2884
62-2 Summary of RTC Standard Static RAM Bank — Indexed Data....2885
62-3 Summary of RTC Extended Static RAM Bank — Indexed Data....2886
62-4 Summary of Sideband Registers—0xD1....2887
63-1 Summary of Memory Mapped I/O Registers—0xFED00000....2905
64-1 Summary of PCI Configuration Registers—0/31/1....2909
64-2 IOxAPIC Interrupt Message Address ....2910
64-3 Summary of Memory-Mapped Registers—Fixed Addresses ....2910
64-4 Summary of IOxAPIC Index/Window-Accessed Registers ....2911
64-5 Summary of I/O Registers—Fixed Location ....2912
65-1 Summary of Sideband Registers, Part 1 of 2—Sideband Port 0xC2....2940
65-2 Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5....2944
67-1 Summary of Sideband Registers—0xD0....3739
68-1 Summary of Sideband Registers (Lane 0 of 19) — 0xBA....3778
68-2 Summary of Sideband Registers (Lane 1 of 19) — 0xBA....3778
68-3 Summary of Sideband Registers (Lane 2 of 19) — 0xBA....3779
68-4 Summary of Sideband Registers (Lane 3 of 19) — 0xBA....3779
68-5 Summary of Sideband Registers (Lane 4 of 19) — 0xBA....3780
68-6 Summary of Sideband Registers (Lane 5 of 19) — 0xBA....3780
68-7 Summary of Sideband Registers (Lane 6 of 19) — 0xBA....3781
68-8 Summary of Sideband Registers (Lane 7 of 19) — 0xBA....3781
68-9 Summary of Sideband Registers (Lane 8 of 19) — 0xBA....3782
68-10 Summary of Sideband Registers (Lane 9 of 19) — 0xBA....3782
68-11 Summary of Sideband Registers (Lane 10 of 19) — 0xBA....3783
68-12 Summary of Sideband Registers (Lane 11 of 19) — 0xBA....3783
68-13 Summary of Sideband Registers (Lane 12 of 19) — 0xBA....3784
68-14 Summary of Sideband Registers (Lane 13 of 19) — 0xBA....3784
68-15 Summary of Sideband Registers (Lane 14 of 19) — 0xBA....3785

68-16 Summary of Sideband Registers (Lane 15 of 19) — 0xBA..... 3785
68-17 Summary of Sideband Registers (Lane 16 of 19) — 0xBB..... 3786
68-18 Summary of Sideband Registers (Lane 17 of 19) — 0xBB..... 3786
68-19 Summary of Sideband Registers (Lane 18 of 19) — 0xBB..... 3787
68-20 Summary of Sideband Registers (Lane 19 of 19) — 0xBB..... 3787
69-1 Model Specific Register (MSR) Index ..... 3796

§ §

# Volume 1: Program Overview

# 1 Introduction and Product Offerings

*Note:* This document contains preliminary information and is highly subject to change.

## 1.1 Overview

The Intel Atom® Processor C3000 Product Family is the next generation System-On-Chip (SoC) built on the latest Intel 14-nanometer process technology. This highly-integrated SoC contains two to 16 next generation 64-bit Intel Atom® processor cores depending on the product SKU. The SoCs were created to address the growing needs for the emerging markets like Microserver, Cloud Storage and growing Communication Infrastructure market segments.

*Note:* Throughout this document, the Intel Atom® Processor C3000 Product Family SoC will be referred to as simply SoC.

Network and Enterprise Storage SKUs are targeted for long life supply and extended communications reliability. Four Extended Temperature SKUs provide operation over an extended temperature range.

The parameter $T_{CASE}$ is mentioned in this chapter and is the temperature of the SoC package measured at the geometric center of the topside of the Integrated Heat Spreader (IHS).

*Note:* The product SKUs, feature lists, and values in the following tables are preliminary based on current expectations and are subject to change without notice:

- Table 1-1, “Features by product SKU - Server and Cloud Storage SKUs 0 through 4”,
- Table 1-2, “Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10”,
- Table 1-3, “Features by Product SKU - Extended Temperature SKUs 11 through 14”

### Table 1-1. Features by product SKU - Server and Cloud Storage SKUs 0 through 4 (Sheet 1 of 2)

| Features | C3955 SKU 0 | C3950 SKU 1 | C3850 SKU 2 | C3830 SKU 3 | C3750 SKU 4 |
|---|---|---|---|---|---|
| Number of 64-bit Intel Atom® Microarchitecture Goldmont Cores | 16 | | 12 | | 8 |
| Number of threads per Core | 1 | | | | |
| Processor base frequency (GHz) | 2.1 | 1.7 | 2.1 | 1.9 | 2.2 |
| Intel® Turbo Boost Technology | Enabled | Enabled | Enabled | Enabled | Enabled |
| P0n Turbo frequency @ 70 °C (GHz) | 2.4 | 2.2 | 2.4 | 2.3 | 2.4 |
| Total SoC L2 Cache (MB) | 16 | | 12 | | 16 |
| L2 Cache for each Core (MB) | 2 (shared by two cores) | | | | 2 (dedicated) |
| Maximum number of memory channels available | 2 | | | | |
| Maximum DDR4 (1.2V) memory data rate supported (MT/s) | 2400 | | | 2133 | 2400 |
| Number of High-Speed I/O (HSIO) Lanes shared between PCIe*, SATA, and USB 3.0 | 20 | | | 12 | |
| HSIO Lane Numbers (0 through 19) available | Lanes 0-19 | | | Any 12 Lanes from Lanes 0-19 | |
| Maximum number of PCI Express* (8.0 / 5.0 / 2.5 GT/s) Lanes | 16 via HSIO Lanes | | | 12 via HSIO Lanes | |
| Maximum number of SATA (6.0 / 3.0 / 1.5 Gbps) Lanes | 16 via HSIO Lanes | | | 12 via HSIO Lanes | |
| Maximum number of sets of USB 3.0 signals SSTX+/SSTX- SSRX+/SSRX- | 4 via HSIO Lanes | | | | |
| Number of sets of USB 2.0 signals (DATA+/DATA-) | 4 | | | | |

**Table 1-1. Features by product SKU - Server and Cloud Storage SKUs 0 through 4 (Sheet 2 of 2)**

| Features | C3955 SKU 0 | C3950 SKU 1 | C3850 SKU 2 | C3830 SKU 3 | C3750 SKU 4 |
|---|---|---|---|---|---|
| LAN Controller 0 (Gb/s) [1] | 10<br>2.5<br>1 | | | | |
| LAN Controller 1 (Gb/s) [1] | 10<br>2.5<br>1 | 1 | not available | | |
| Intel® Management Engine (Intel® ME) | Yes | Yes | | | |
| Innovation Engine (IE) | Yes | Yes | | | |
| Intel® QuickAssist Technology (Intel® QAT) | No | | | | |
| Intel® QAT Speed (MHz) | n/a | | | | |
| Intel® QAT Speed | n/a | | | | |
| Intel® QAT Cryptographic Functions | n/a | | | | |
| Relative temperature offset ($T_{CONTROL}$) from the Thermal Monitor Trip Temperature at which fans will be engaged (°C) | 15 | | | | |

***Notes:***
1. Each LAN controller must run the same LEK. See Chapter 13, "LAN Controllers" for details.

## Table 1-2. Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10 (Sheet 1 of 2)

| Features | C3958 SKU 5 | C3858 SKU 6 | C3758 SKU 7 | C3558 SKU 8 | C3538 SKU 9 | C3338 SKU 10 |
|---|---|---|---|---|---|---|
| Number of 64-bit Intel Atom® Microarchitecture Goldmont Cores | 16 | 12 | 8 | 4 | | 2 |
| Number of threads per Core | 1 | | | | | |
| Processor base frequency (GHz) | 2 | 2 | 2.2 | 2.2 | 2.1 | 1.5 |
| Intel® Turbo Boost Technology | No | | | | | Yes |
| P0n Turbo frequency @ 70°C (GHz) | N/A | | | | | 2.2 |
| Total SoC L2 Cache (MB) | 16 | 12 | 16 | 8 | | 4 |
| L2 Cache for each Core (MB) | 2 (shared by two cores) | | 2 (dedicated) | | | |
| Maximum number of memory channels available | 2 | | | | | 1 [3] |
| Maximum DDR4 (1.2V) memory data rate supported (MT/s) | 2400 | | | 2133 | | 1866 |
| Number of High-Speed I/O (HSIO) Lanes shared between PCIe*, SATA, and USB 3.0 | 20 | | | 12 | | 10 [1] |
| HSIO Lane Numbers (0 through 19) available | Lanes 0-19 | | | Any 12 Lanes from Lanes 0-19 | | Any 10 Lanes from Lanes 0-19 |
| Maximum number of PCI Express* (8.0 / 5.0 / 2.5 GT/s) Lanes | 16 via HSIO Lanes | | | 12 via HSIO Lanes | | 10 via HSIO Lanes |
| Maximum number of SATA (6.0 / 3.0 / 1.5 Gbps) Lanes | 16 via HSIO Lanes | | | 12 via HSIO Lanes | | 10 via HSIO Lanes |
| Maximum number of sets of USB 3.0 signals SSTX+/SSTX- SSRX+/SSRX- | 4 via HSIO Lanes | | | | | |
| Number of sets of USB 2.0 signals (DATA+/DATA-) | 4 | | | | | |

**Table 1-2. Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10 (Sheet 2 of 2)**

| Features | C3958 SKU 5 | C3858 SKU 6 | C3758 SKU 7 | C3558 SKU 8 | C3538 SKU 9 | C3338 SKU 10 |
|---|---|---|---|---|---|---|
| LAN Controller 0 (Gb/s) [2] | 10<br>2.5<br>1 | | | | | 2.5<br>1 |
| LAN Controller 1 (Gb/s) [2] | 10<br>2.5<br>1 | | | 2.5<br>1 | | |
| Intel® Management Engine (Intel® ME) | Yes | | | | | |
| Innovation Engine (IE) | Yes | | | | | |
| Intel® QuickAssist Technology (Intel® QAT) | Yes | | | | | No |
| Intel® QAT Speed | High | High | Medium | Low | Low | n/a |
| Intel® QAT Cryptographic Functions | Yes | | | | No | n/a |
| Intel® QAT Compression/ Decompression | Yes | | | | | n/a |
| Relative temperature offset ($T_{CONTROL}$) from the Thermal Monitor Trip Temperature at which fans will be engaged (°C) | 20 | | | | | |

***Notes:***
1. This product SKU has 10 total available HSIO Lanes. To meet the specified TDP, four of the 10 HSIO Lanes must be disabled with Soft Straps or BIOS. If more than six HSIO Lanes are enabled, the TDP will increase with a thermal resistance between the processor's case and the ambient air. An adequate thermal solution must be designed if more than 6 lanes are used.
2. Each LAN controller must run the same LEK. See Chapter 13, "LAN Controllers" for details.
3. Only one memory channel may be used with this SKU. It can be memory channel 0 or memory channel 1.

### Table 1-3. Features by Product SKU - Extended Temperature SKUs 11 through 14 (Sheet 1 of 2)

| Features | C3808 SKU 11 | C3708 SKU 12 | C3508 SKU 13 | C3308 SKU 14 |
|---|---|---|---|---|
| Number of 64-bit Intel Atom® Microarchitecture Goldmont Cores | 12 | 8 | 4 | 2 |
| Number of threads per Core | 1 | | | |
| Processor base frequency (GHz) | 2 | 1.7 | 1.6 | 1.6 |
| Intel® Turbo Boost Technology | No | | | Yes |
| P0n Turbo frequency @ 70°C (GHz) | N/A | | | 2.1 |
| Total SoC L2 Cache (MB) | 12 | 16 | 8 | 4 |
| L2 Cache for each Core (MB) | 2 (shared by two cores) | 2 (dedicated) | | |
| Maximum number of memory channels available | 2 | | | 1 [2] |
| Maximum DDR4 (1.2V) memory data rate supported (MT/s) | 2133 | | 1866 | |
| Number of High-Speed I/O (HSIO) Lanes shared between PCIe*, SATA, and USB 3.0 | 20 | | 8 | 6 |
| HSIO Lane Numbers (0 through 19) available | Lanes 0-19 | | Any 8 Lanes from Lanes 0-19 | Any 6 Lanes from Lanes 0-19 |
| Maximum number of PCI Express* (8.0 / 5.0 / 2.5 GT/s) Lanes | 16 via HSIO Lanes | | 8 via HSIO Lanes | 6 via HSIO Lanes |
| Maximum number of SATA (6.0 / 3.0 / 1.5 Gbps) Lanes | 16 via HSIO Lanes | | 8 via HSIO Lanes | 6 via HSIO Lanes |
| Maximum number of sets of USB 3.0 signals SSTX+/SSTX- SSRX+/SSRX- | 4 via HSIO Lanes | | | |
| Number of sets of USB 2.0 signals (DATA+/DATA-) | 4 | | | |

## Table 1-3. Features by Product SKU - Extended Temperature SKUs 11 through 14 (Sheet 2 of 2)

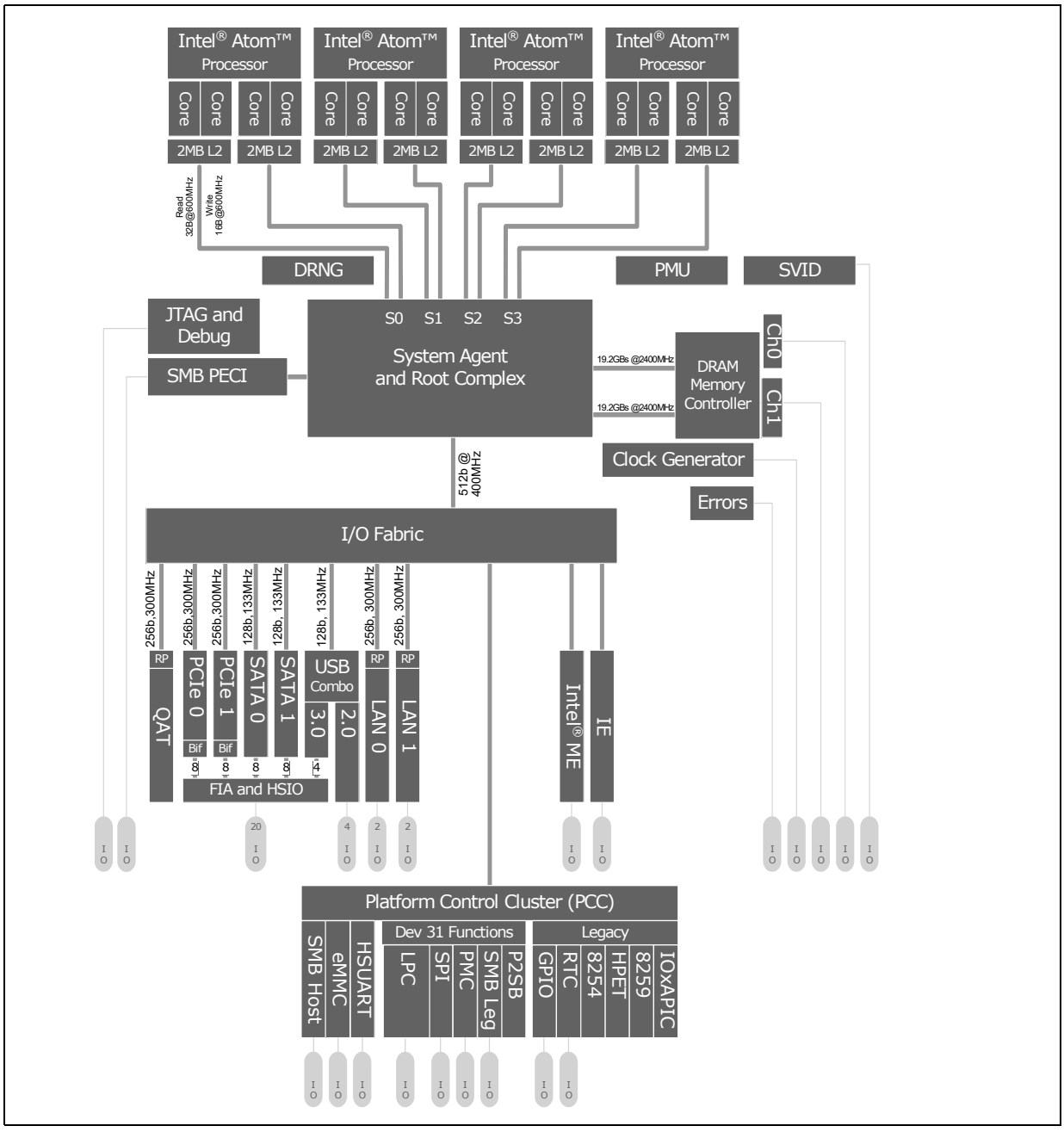
| Features | C3808 SKU 11 | C3708 SKU 12 | C3508 SKU 13 | C3308 SKU 14 |
|---|---|---|---|---|
| LAN Controller 0 (Gb/s) [1] | 10<br>2.5<br>1 | | 2.5<br>1 | |
| LAN Controller 1 (Gb/s) [1] | 10<br>2.5<br>1 | | 2.5<br>1 | |
| Intel® Management Engine (Intel® ME) | Yes | | Yes | |
| Innovation Engine (IE) | Yes | | Yes | |
| Intel® QuickAssist Technology (Intel® QAT) | Yes | | | |
| Intel® QAT Speed | High | Medium | Low | |
| Intel® QAT Cryptographic Functions | Yes | | | |
| Intel® QAT Compression/ Decompression | Yes | | | |
| Relative temperature offset ($T_{CONTROL}$) from the Thermal Monitor Trip Temperature at which fans will be engaged (°C) | 20 | | | |

***Notes:***
1. Each LAN controller must run the same LEK. See Chapter 13, "LAN Controllers" for details.
2. Only one memory channel may be used with this SKU. It can be memory channel 0 or memory channel 1.

## 1.2 SoC Key Features

This section describes the features offered in the SoC.

**Figure 1-1. High-Level Block Diagram**



### 1.2.1 Processor Details

The SoC has up to sixteen 64-bit Intel Atom® processor cores:

- Each core has an L1 (first-level) cache consisting of:
  - 24-KB data, parity protected.
  - 32-KB instruction, parity protected.
- L2 (second-level) instruction/data cache, ECC protected:
  - SoC with 2, 4 or 8 cores have a 2-MB L2 cache per core.
  - SoC with 12 or 16 cores share a 2-MB L2 cache per 2 cores.

### 1.2.2 Supported Technologies

- Intel® Virtualization Technology (Intel® VT) for Intel® 64 and IA-32 Intel® Architecture (Intel® VT-x).
- Intel® Virtualization Technology (Intel® VT-x) for Directed I/O (Intel® VT-d).
- Advanced Encryption Standard New Instructions (AES-NI).
- Secure Hash Algorithm New Instruction (SHA-NI).
- Intel® 64 Architecture.
- Intel® Streaming SIMD Extensions 4.1 (Intel® SSE4.1).
- Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2).
- Execute Disable Bit.
- Intel® Turbo Boost Technology.
- Enhanced Intel SpeedStep® Technology.

### 1.2.3 System Memory Support

- Up to two 32/40 bit [1] or 64/72 bit wide memory channels. Up to two DIMMs per channel.
- Single-rank or dual-rank per DIMM module.
- ECC [4] and non-ECC support.
- DDR4 (1.2V):
  - 1600, 1866 [2], 2133, and 2400 MT/s.
  - x8 or x16 width for unbuffered memory (UDIMM, SO-DIMM and memory down).
  - x4 or x8 width devices for RDIMM
  - Minimum memory is:
    32 bit [1] wide memory channel: 1GB (4 Gb, 1 rank, x16 width)
    64-bit wide memory channel: 2GB (4 Gb, 1 rank, x16 width).
  - Maximum memory single channel is 128 GB (16 Gb, x4 width, 4 ranks, 64-bit).
  - Both memory channels maximum is 256 GB (128GB with two 64-bit channels).
- SoC memory population rules:
  - Both channels must be 32 bit [1] or 64 bits wide
  - At least 1 memory channel must be populated.
    Either channel 0 or channel 1 maybe disabled.
  - Both channels require either unbuffered memory (UDIMM, SO-DIMM or memory down), or registered memory (RDIMM and/or NVRDIMM). Technology mixing is not supported.
  - Rank mixing is allowed but can result in a speed bin loss.
  - Each populated channel must be x4 or x8 or x16 only (no width mixing within a channel) 4 Gb, 8 Gb, and 16 Gb technology device mixing between channels is allowed (i.e., Channel 0 can be x4 while Channel 1 is x8).

***Notes:***

1. A0 and A1 step silicon only support 64/72-bit wide memory channels.
2. The silicon design supports 1866 MT/s but is not validated on Intel platforms. Customers are responsible for validation.
3. With a standard DIMM and NVDIMM in the same channel, the NVDIMM must be in the second slot or be the same size or smaller than the standard DIMM.
4. The ECC SDRAM device must be of the same type and width as the SDRAM devices used for the data bits. This limitation is driven by the DDR4 spec.
   - It is NOT alright to use x16 SDRAM devices for data and a x8 device for ECC.
   - It is NOT alright to use x8 SDRAM devices for data and a x4 device for ECC.
5. Memory based on 16 Gb dual rank technology (including maximum memory configuration of 256GB) is supported by silicon but not validated on Intel platforms.

### 1.2.4 Integrated Clock Generator

The integrated clock generator requires two external crystals (32.768-kHz and 25-MHz) and eliminates the need for an external clock chip.

- Generates all required internal clocks.
- Two flex clock outputs for platform use.
- Five PCIe output clocks.
- Optional Spread-Spectrum Clocking support.

### 1.2.5 Intel® QuickAssist Technology (Intel® QAT)

Intel® QuickAssist Technology is only available on Network and Enterprise Storage SKUs and provides hardware acceleration for:

Symmetric cryptographic functions:

- Cipher operations.
- Hash/authenticate operation.
- Cipher-hash combined operation.

Public key functions:

- RSA operation, encryption/decryption and digital signature generation/verification.
- Modular exponentiation for Diffie-Hellman (DH).
- DSA parameter generation and digital signature generation/verification.
- Key derivation operation.
- Elliptic curve cryptography: ECDSA and ECDH.

Compression/Decompression:

- Deflate.

### 1.2.6 Intel® Management Engine (Intel® ME)

The Intel® ME provides the services that are released and documented in Intel® Server Platform Services Firmware (SPS 4.0) release.

### 1.2.7 Innovation Engine (IE)

The Innovation Engine (IE) is a complete, embedded, Intel Architecture system intended for Intel customers (system builders) to create innovative or differentiating firmware of their choosing. The IE is available on all SoC product SKUs bit is not required for normal system operation.

The IE is based on Intel® Quark™ technology:

- 32-bit Intel Architecture processor core
- Suite of connections to a number of platform resources

For security, the IE software is cryptographically bound to the owner, the system builder, using a digital-signing technique.

For the IE, the system builder can install an embedded operating system, drivers and application they develop on their own, or purchase them from a third-party vendor. Intel does not provide operating systems, drivers or applications for the IE.

The IE usage is only limited by its technical capabilities. Sample uses include using the IE as a replacement for a low-end Baseboard Management Controller (BMC), or offloading fast-loop sensor monitoring from the BMC.

## 1.2.8 Interface Features

### 1.2.8.1 High Speed I/O Lanes

Depending on the SoC product SKU, there are up to 20 High-Speed I/O (HSIO) lanes that are shared between the PCI Express*, Serial ATA (SATA) and Universal Serial Bus (USB) 3.0 ports. See the Chapter 10, "Flexible I/O Adapter (FIA) Overview" for more information on the sharing. The maximum number of available lanes is shown in Table 1-1 on page 107, Table 1-2 on page 109, and Table 1-3 on page 111.

#### 1.2.8.1.1 PCIe Root Port Clusters

- Up to two PCIe Root Port Clusters 0 and 1.
- Fully-compliant to the *PCI Express Base Specification Revision 3.0*.
- Each 1x8 PCIe Root Port Cluster maybe bifurcated by BIOS to:
  - 1x8
  - 2x4
  - 2x2 and 1x4
  - 4x2
  - 4x1
- Minimum Root Port width is 1-lane.
- Maximum Root Port width is 8-lanes with degrades to x4, x2 or x1.
- Lane Reversal supported.
- Polarity Inversion supported.
- No support for peer-to-peer transfers between PCIe Root Ports.

#### 1.2.8.1.2 Serial ATA (SATA)

The SATA ports are 3.1 or 2.0 capable:

- Up to 16 SATA ports:
  - Implemented with up to two controllers.
  - Up to eight ports per controller.
  - One SGPIO per controller.
- No support for IDE mode.

#### 1.2.8.1.3 Universal Serial Bus (USB)

There is a single combo USB 3.0 and USB 2.0 controller with support for:

- Up to four USB 3.0 compliant ports.
- USB 2.0 ports have dedicated SoC I/O pins.
- USB 3.0 ports share HSIO lanes as described above.

### 1.2.8.2 LAN Ports

Depending on the SoC product SKU, there is either one or two, dual-port, Enterprise-Class, LAN controllers. The controllers support 10GbE or 2500BASE-X depending on the product SKU.

- Based on Intel® Ethernet Controller X55x
- 10GbE LAN ports support SFI, KR, 2500BASE-X, SGMII and 1000BASE-KX
- 2500BASE-X LAN ports support 2500BASE-X, SGMII and 1000BASE-KX
- SGMII is supported for Full Duplex 10Mb, 100Mb and 1Gb
- No support for Half Duplex at 10Mb, 100Mb, 1Gb, 2.5Gb or 10Gb
- Management Interface via NC-SI or SMBus

### 1.2.8.3 SMBus Controllers

*Note:* There are seven sets of SMBus pins. And so a maximum of seven SMBus controllers may be configured to be active.

The SoC has the following 10 SMBus controllers:

- Legacy and Host SMBus controllers used by the IA cores
- PECI over SMBus controller
- LAN Management SMBus controller for BMC use
- Three Management Engine SMBus controllers
- Three Innovation Engine SMBus controllers

### 1.2.8.4 Embedded Multi Media Card (eMMC)

The Embedded Multi-Media Controller (eMMC) is a high speed Flash memory controller that provides low cost, small form factor storage.

- Support for eMMC v5.0
- Eight-bit interface maximum frequency of 200 MHz for transfer rates up to 400 MB/s

### 1.2.8.5 High-Speed Universal Asynchronous Receiver/Transmitter

*Note:* At most, three of the SoC four High Speed Universal Asynchronous Receiver/Transmitter (HSUART) controllers can be active. One set of the SoC HSUART pins is configurable to be either the Host Root HSUART or else the Innovation Engine (IE) HSUART.

There are 4 HSUART each supporting Clear-To-Send (CTS) and Request-To-Send (RTS) depending on MUXing. Three (3) of the HSUARTs are in the IA Host space and 1 is in the Innovation Engine Root space.

- Baud rate from 300 bps to 3.6864 Mbps

#### 1.2.8.6 Low Pin Count (LPC)

- LPC
  - Interface is described in the *Intel® Low Pin Count (LPC) Interface Specification*
  - Support for Security Device Trusted Platform Module (TPM)

#### 1.2.8.7 Serial Peripheral Interface (SPI)

- Supports up to two SPI NOR Flash devices and a SPI NOR TPM device
- Supports 17 MHz, 30 MHz, and 48 MHz SPI Devices
- Supports Serial Flash Discovery Parameters (SFDP)
- Supports Quad I/O Fast Read, Quad Output Fast Read, Dual IO Fast Read

### 1.2.9 Package

The SoC is available in one package size:

- A 34 x 28 mm 1310-ball FCBGA package (FCBGA15)
- Integrated Heat Spreader (IHS)
- 0.65 mm minimum outer pin pitch, maintaining 0.7mm pitch for breakout direction to keep 0.7 mm pitch board routing rules.

### 1.2.10 Testability

- Includes boundary-scan for board and system level testability.

## 1.3 Terminology

### Table 1-4. Terminology (Sheet 1 of 6)

| Term | Description |
|---|---|
| 1000BASE-KX | 1000BASE-KX is the IEEE 802.3ap electrical specification for transmission of 1 Gb/s Ethernet over the backplane. |
| 2500BASE-X | Refer to 2.5 GbE in this table. Note this is not an IEEE Standard. |
| 1000BASE-T | 1000BASE-T is the specification for 1 Gb/s Ethernet over category 5e twisted pair cables as defined in IEEE 802.3 clause 40. |
| 2.5 GbE | The 2.5-GbE link mode is a special enhanced-speed mode of the 1000BASE-X link mode. This mode should not be confused with third-party solutions being offered in the marketplace today. Contact your local Intel representative for further information on the scope of this feature. |
| ACPI | Advanced Configuration and Power Interface |
| ADR | Asynchronous DRAM Refresh |
| AEN | Asynchronous Event Notification |
| AFE | Analog Front End |
| Aggressor | A network that transmits a coupled signal to another network. |
| AH | IP Authentication Header - An IPsec header providing authentication capabilities defined in RFC 4302. |
| Anti-Etch | Any plane-split, void, or cutout in a VCC or GND plane is referred to as an anti-etch. |
| ARP | Address Resolution Protocol |
| ASF | Alert Standard Format |
| b/w | Bandwidth |
| BER | Bit Error Rate |
| BGA | Ball Grid Array |
| BIOS | Basic Input/Output System. BIOS is the same as what is called UEFI IA Firmware in this document. Both terms are used and refer to the same thing. |
| BIST FIS | Built-In Self-Test Frame Information Structure |
| BMC | Baseboard Management Controller (often used interchangeably with MC) |
| BT | Bit Time |
| Bus Agent | A component which represents a load on a bus. |
| CDV | Cell Delay Variation |
| CMC | Common Mode Choke |
| CMCI | Corrected Machine Check error Interrupt |
| CMP | Chip-Level Multi Processing (CMP) |
| CRC | Cyclic Redundancy Check |
| CSR | Configuration and Status Register |
| DCA | Direct Cache Access |
| DDR3 | Third generation Double Data Rate SDRAM memory technology that is the successor to DDR2 SDRAM. |
| DDR4 | Forth generation Double Data Rate SDRAM memory technology that is the successor to DDR3 SDRAM. |
| DFT | Design for Testability |
| DMA | Direct Memory Access |
| DMTF | Distributed Management Task Force standard body |
| DP | Display Port |

## Table 1-4. Terminology (Sheet 2 of 6)

| Term | Description |
|---|---|
| DQ | Descriptor Queue |
| DUT | Device Under Test |
| DW | Double Word (4 bytes) |
| ECC | Error Correction Code |
| EEE | Energy Efficient Ethernet - IEEE 802.3az standard |
| EEPROM | Electrically Erasable Programmable Memory. A non-volatile memory directly accessible from the host. |
| EHCI | Enhanced Host Controller Interface |
| EMI | Electromagnetic Interference |
| Enhanced Intel SpeedStep® Technology | Allows the operating system to reduce power consumption when performance is not needed. |
| EOP | End of Packet |
| EPT | Extended Page Tables |
| ESD | Electrostatic Discharge |
| Execute Disable Bit | The Execute Disable bit allows memory to be marked as executable or non-executable, when combined with a supporting operating system. If code attempts to run in non-executable memory the processor raises an error to the operating system. This feature prevents some classes of viruses or worms that exploit buffer overrun vulnerabilities and improves the overall system security. See the Intel® 64 and IA-32 architectures software developer manuals for more detailed information. |
| FC | Flow Control |
| FCS | Frame Check Sequence |
| Firmware (FW) | Embedded code on the LAN controller that implements the NC-SI protocol and pass-through functionality. |
| Flight Time | A term in the timing equation that includes the signal propagation delay, any effects the system has on the driver TCO, plus any adjustments to the signal at the receiver needed to guarantee the receiver setup time. More precisely, flight time is defined as the following:<br>• The time difference between a signal at the input pin of a receiving agent crossing the switching voltage (adjusted to meet the receiver manufacturer conditions required for AC timing specifications; i.e., ringback, etc.) and the output pin of the driving agent crossing the switching voltage when the driver is driving a test load used to specify the driver AC timings.<br>Maximum and Minimum Flight Time – Flight time variations are caused by many different parameters. The more obvious causes include the board dielectric constant variation, changes in load condition, crosstalk, power noise, variation in termination resistance, and differences in I/O buffer performance as a function of temperature, voltage, and manufacturing process. Some less obvious causes include effects of Simultaneous Switching Output (SSO) and packaging effects.<br>Maximum flight time is the largest acceptable flight time a network experiences under all conditions.<br>Minimum flight time is the smallest acceptable flight time a network experiences under all conditions. |
| FS | Full-speed. Refers to USB. |
| Host Interface | RAM on the LAN controller shared between the firmware and the host, and used to pass commands from the host to the firmware and responses from the firmware to the host. |
| HPC | High Performance Computing |
| HS | High-speed. Refers to USB. |
| IIO | The Integrated I/O Controller. An I/O controller that is integrated in the processor die. |
| IOMMU | Input/Output Memory Management Unit |
| iMC | The Integrated Memory Controller. A Memory Controller that is integrated in the processor die. |
| IE | Innovation Engine |
| Intel® 64 Technology | 64-bit memory extensions to the IA-32 architecture. Further details on the Intel 64 architecture and the programming model are found at: http://developer.intel.com/technology/intel64/. |
| Intel® ME | Intel® Management Engine |

## Table 1-4. Terminology (Sheet 3 of 6)

| Term | Description |
|---|---|
| IPC | Inter-Processor Communication |
| IPDC | Internet Portal Data Center |
| IPI | Inter-Processor Interrupt |
| IPG | Inter-Packet Gap |
| IPMI | Intelligent Platform Management Interface specification |
| Intel® QuickAssist Technology | Intel® QuickAssist Technology consists of a family of interrelated Intel and industry-standard technologies that simplify the use and deployment of accelerators on the Intel platforms. |
| Intel® Virtualization Technology (Intel® VT) | Processor virtualization which when used in conjunction with the Virtual Machine Monitor software enables multiple, robust independent software environments inside a single platform. |
| Integrated Heat Spreader (IHS) | A component of the processor package used to enhance the thermal performance of the package. Component thermal solutions interface with the processor at the IHS surface. |
| ISA | Instruction Set Architecture |
| ISI | The effect of a previous signal (or transition) on the interconnect delay. For example, when a signal is transmitted down a line and the reflections due to the transition have not completely dissipated, the following data transition launched onto the bus is affected. Interconnect Symbolic Interference (ISI) is dependent upon frequency, time delay of the line, and the reflection coefficient at the driver and receiver. ISI impacts both timing and signal integrity. |
| Jitter | Any timing variation of a transition edge or edges from the defined Unit Interval (UI). |
| LAN (Auxiliary Power-Up) | If connecting the LAN controller to a power source (occurs even before system power-up). |
| Land | The contact point of the SoC FCBGA package. |
| LCD | Name for the external LAN Connected Device, sometimes also called a PHY device. The LCD name distinguishes between the whole device and the PHY portion within the LCD. |
| LCI | LAN Connect Interface |
| LLC | Last Level Cache |
| LLDP | Link Layer Discovery Protocol defined in IEEE 802.1AB and used by IEEE 802.3az (EEE) for system wake time negotiation. |
| LOM | LAN on Motherboard |
| LPC | Low Pin Count |
| LPI | Low-Power Idle is the low-power state of Ethernet link as defined in IEEE 802.3az. |
| LRU | Least Recently Used. A term used in conjunction with cache hierarchy. |
| LS | Low-speed. Refers to USB. |
| LSO | Large Send Offload |
| LTR | Latency Tolerance Reporting (PCI Express* (PCIe*) protocol) |
| LVR | Linear Voltage Regulator |
| M0 | Intel® Management Engine (Intel® ME) power state where all HW power planes are activated. Host power state is S0. |
| M3 | Intel® Management Engine (Intel® ME) power state where all HW power planes are activated, however the host power state is different than S0 (some host power planes are not activated). Host PCI-E* interface are unavailable to the host SW. Main memory is not available for Intel® Management Engine use. |
| MAC | Media Access Control |
| MC | Management Controller |
| MCA | Machine Check Abort |
| MCTP | DMTF Management Component Transport Protocol (MCTP) specification. A transport protocol to allow communication between a management controller and controlled device over various transports. |
| MDIO | Management Data Input/Output Interface over MDC/MDIO lines |

## Table 1-4. Terminology (Sheet 4 of 6)

| Term | Description |
|---|---|
| ME | See Intel® Management Engine (Intel® ME) |
| MIFS/MIPG | Minimum Inter-Frame Spacing/Minimum Inter-Packet Gap |
| MLC | Mid-Level Cache |
| MMC | Multi-Node Management Controller |
| MMW | Maximum Memory Window |
| MPS | Maximum Payload Size in PCIe specification |
| MSS | Maximum Segment Size. Largest amount of data, in a packet (without headers) that can be transmitted. Specified in bytes. |
| MTU | Maximum Transmit Unit. Largest packet size (headers and data) that can be transmitted. Specified in bytes. |
| NC | Network Controller |
| NC-SI | Network Controller Sideband Interface DMTF Specification |
| Network | The trace of a Printed Circuit Board (PCB) that completes an electrical connection between two or more components. |
| NIC | Network Interface Controller |
| NVDIMM | Nonvolatile DIMM is a DIMM technology that will save its DRAM content to flash on power loss and allow recover of the data on the next system boot. |
| OOB | Out of Band |
| Overshoot | The maximum voltage observed for a signal at the device pad, measured with respect to $V_{CC}$. |
| Pad | The electrical contact point of a semiconductor die to the package substrate. A pad is only observable in simulations. |
| PCI, PCIe* | Peripheral Component Interconnect (Express). In this document, this interconnect refers to the PCI logical layer used by the SoC I/O fabric protocol. |
| PCI Express* | PCI Express |
| Pcode | Power Management Unit Micro-code |
| PCS | Physical Coding Sub layer |
| PEC | Packet Error Code |
| PECI | Platform Environmental Control Interface |
| PHY | Physical Layer Device |
| Pin | The contact point of a component package to the traces on a substrate, such as the motherboard. Signal quality and timings are measured at the pin. |
| PLC | Platform LAN Connect |
| PMA | Physical Medium Attachment |
| PMC | Power Management Controller |
| PMD | Physical Medium Dependent |
| PMU | Power Management Unit |
| Power-Good | Power-Good, or PWRGOOD (an active high signal) indicates that all the system power supplies and clocks are stable. PWRGOOD does go active a predetermined time after system voltages are stable and does go inactive as soon as any of these voltages fail their specifications. |
| Processor | The 64-bit, single-core or multi-core component (package) |
| Processor Core | The term Processor Core refers to the silicon die itself which contains multiple execution cores. Each execution core has an instruction cache, data cache, and shares its 1-MB L2 cache with a sibling execution core. |
| PXE | Preboot Execution Environment |
| Rank | A unit of DRAM corresponding 4 to 16 devices in parallel, ignoring ECC. These devices are usually, but not always, mounted on a DIMM Raw Card. |

## Table 1-4. Terminology (Sheet 5 of 6)

| Term | Description |
|---|---|
| Ringback | The voltage to which a signal changes after reaching its maximum absolute value. Ringback is caused by reflections, driver oscillations, or other transmission line phenomena. |
| RP | Root Port |
| RTC | Real Time Clock |
| SA | System Agent |
| SATA | Serial ATA |
| SDP | Software Defined Pins |
| SerDes | Serializer/Deserializer: A transceiver that converts parallel data to serial data and vice-versa. |
| SFD | Start Frame Delimiter |
| SFI | SerDes Frame Interface |
| SFP | Small Form-factor Pluggable |
| SFP+ | enhanced Small Form-factor Pluggable |
| SGMII | Serialized Gigabit Media Independent Interface<br><br>***Note:*** SoC SGMII is supported for Full Duplex 10mb, 100mb and 1Gb only (No Half Duplex support). |
| SIMD | Single Instruction Multiple Data |
| SMBus | System Management Bus. A two-wire interface through which simple system and power management related devices communicate with the rest of the system. A bus carrying various manageability components, including the LAN controller, BIOS, sensors, and remote-control devices. SMBus is based on the operational principles of the $I^2C$ two-wire serial bus from Philips Semiconductor. |
| So-DIMM | Small outline Dual In-line Memory Module |
| SPD | Serial Presence Detect |
| SPI | Serial Peripheral Interface |
| SSE | Intel® Streaming SIMD Extensions (Intel® SSE) |
| SSO | Simultaneous Switching Output (SSO) effects are differences in electrical timing parameters and degradation in signal quality caused by multiple signal outputs simultaneously switching voltage levels in the opposite direction from a single signal or in the same direction. These are called odd mode and even mode switching, respectively. This simultaneous switching of multiple outputs creates higher current swings that may cause additional propagation delay (“push-out”) or a decrease in propagation delay (“pull-in”). These SSO effects may impact the setup and/or hold times and are not always taken into account by simulations. System timing budgets include margin for SSO effects. |
| STR | Suspend To RAM |
| Stub | The branch from the bus trunk terminating at the pad of an agent. |
| SVR | Switched Voltage Regulator |
| TAC | Thermal Averaging Constant |
| TCO | Total Cost of Ownership (TCO) System Management |
| TDP | Thermal Design Power |
| TDR | Time Domain Reflectometry |
| TLB | Translation Lookaside Buffer |
| TLP | Transaction Layer Packet in the PCI Express specification |
| Trunk | The main connection, excluding interconnect branches, from one end-agent pad to the other end-agent pad. |
| TSO | Transmit Segmentation Offload. A mode in which a large TCP/UDP I/O is handed to the device which the device then segments into L2 packets according to the requested MSS. |
| TSOD | Temperature Sensor On DIMM |

## Table 1-4. Terminology (Sheet 6 of 6)

| Term | Description |
|---|---|
| UDIMM | Unbuffered Dual In-line Memory Module |
| UEFI IA Firmware | This is another term used for BIOS. Both terms are used and refer to the same thing. |
| UHCI | Universal Host Controller Interface |
| Undershoot | The minimum voltage extending below $V_{SS}$ observed for a signal at the device pad. |
| Unit Interval (UI) | Binary signaling convention that is measure of time representing the transmission of a single data bit in a serial data stream. One bit is sent for every forwarded clock edge, whether it be a rising edge or a falling edge. If a number of edges are collected at instances t1, t2, tn,...., tk, then the UI at instance "n" is defined as the following: UI n = t n - t n-1. |
| USB | Universal Serial Bus |
| VCC | Processor core-power supply |
| Vcc_core | The core-power rail for the processor. |
| Victim | A network that receives a coupled crosstalk signal from another network is called the victim network. |
| VLAN | Virtual LAN |
| VPD | Vital Product Data (PCI protocol) |
| VRD | The Voltage Regulator Down (a down on the board solution) for the processor. The VRD is a DC-DC converter module that supplies the required voltage and current to a single processor. |
| VSS | Processor ground |
| Intel® VT-d | Intel® Virtualization Technology (Intel® VT-x) for Directed I/O (Intel® VT-d) |
| Intel® VT-x | Intel® Virtualization Technology (Intel® VT) for Intel® 64 and IA-32 Intel® Architecture (Intel® VT-x) |
| x1 | Refers to a link or port with one physical lane. |
| x2 | Refers to a link or port with two physical lanes. |
| x4 | Refers to a link or port with four physical lanes. |
| x8 | Refers to a link or port with eight physical lanes. |
| XDP | eXtended Debug Port |

## 1.4 Related Documents

Refer to the following documents for additional information and guidance in developing platform boards using the SoC.

**Table 1-5. Reference Documents (Sheet 1 of 3)**

| Document | Document Number / Location |
|---|---|
| *Intel® 64 and IA-32 Architectures Software Developer's Manual* | 325462 |

## Table 1-5. Reference Documents (Sheet 2 of 3)

| Document | Document Number / Location |
|---|---|
| *Intel® Low Pin Count (LPC) Interface Specification* | 251289 |
| *PCI Express Base Specification Revision 3.0* | https://pcisig.com/specifications |
| *PCI Local Bus Specification, Revision 3.0* | https://pcisig.com/specifications |
| *PCI Express Card Electromechanical Specification, Revision 3.0* | https://pcisig.com/specifications |
| *PCI Express Mini Card Electromechanical Specification, Revision 2.0* | https://pcisig.com/specifications |
| *PCI-to-PCI Bridge Architecture Specification, Revision 1.2* | https://pcisig.com/specifications |
| *PCI Bus Power Management Interface Specification, Revision 1.2* | https://pcisig.com/specifications |
| *Universal Serial Bus 3.0 Specification, Revision 1.0* | http://www.usb.org/developers/docs/ |
| *Universal Serial Bus Specification, Revision 2.0* | http://www.usb.org/developers/docs/usb20_docs/ |
| *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1* | https://www.intel.com/content/www/us/en/io/universal-serial-bus/extensible-host-controler-interface-usb-xhci.html |
| *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* | https://www.intel.com/content/www/us/en/io/universal-serial-bus/ehci-specification-for-usb.html |
| *System Management Bus (SMBus) Specification, Version 3.0* | http://smbus.org/specs/SMBus_3_0_20141220.pdf |
| *Alert Standard Format (ASF) Specification, Version 2.0* | https://www.dmtf.org/sites/default/files/standards/documents/DSP0136.pdf |
| *Management Component Transport Protocol (MCTP) Base Specification, Version 1.1.0* | https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.1.0.pdf |
| *Intel® QuickAssist Technology API Programmer's Guide* | https://01.org/sites/default/files/page/330684-001us_api_pg.pdf |
| *Advanced Configuration and Power Interface Specification Version 5.0, Errata A* | http://uefi.org/specifications |
| *SPD Annex L, Serial Presence Detect (SPD) for DDR4 SDRAM Modules* | https://www.jedec.org/ *JDEC Standard Number* 21-C |
| *PCI-SIG* SR-IOV Primer: An Introduction to SR-IOV Technology* | https://www.intel.com/content/dam/doc/application-note/pci-sig-sr-iov-primer-sr-iov-technology-paper.pdf |
| *Embedded Multi-Media Card (eMMC), Electrical Standard* | https://www.jedec.org/document_search?search_api_views_fulltext=jesd84-b51 |

## Table 1-5. Reference Documents (Sheet 3 of 3)

| Document | Document Number / Location |
|---|---|
| *PC16550D Universal Asynchronous Receiver/Transmitter With FIFOs (Rev. C)* | http://www.ti.com/lit/ds/symlink/pc16550d.pdf |
| *Serial Flash Discoverable Parameters (SFDP), JEDEC Specification JESD216B* | https://www.jedec.org/standards-documents/docs/jesd216b |
| *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a* | https://www.intel.com/content/dam/www/public/us/en/documents/technical-specifications/software-developers-hpet-spec-1-0a.pdf |

***Note:*** For release dates or latest revisions and documentation numbers, contact the appropriate Intel field representative.

§ §

# Volume 2: Functional



February 2018
Order Number: 337018-002

# 2 Multi-Core Intel Atom® Processors

***Warning:*** This document contains preliminary information and is highly subject to change.

## 2.1 Introduction

The SoC is a highly-integrated System-on-Chip (SoC). The SoC core is a 3-way superscalar, out-of-order, IA core. It is designed using the cutting edge 14 nm process. The SoC core provides roughly 40% Instructions-Per-Cycle (IPC) increase over the previous Intel Atom® processor core.

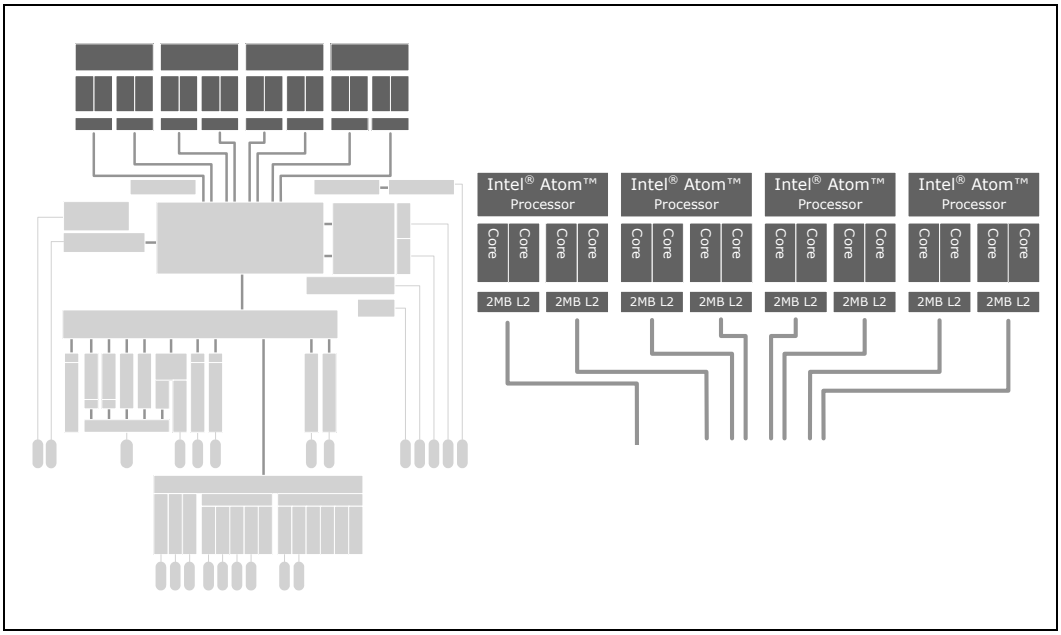
Also see Table 1-5, "Reference Documents" for newest version of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

The SoC supports a variety of SKUs: up to 16 cores (with SKUs containing 2, 4, 8, 12 or 16 cores). Refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The cores and the rest of the SoC are shown in a general block diagram in Figure 2-1. Most of the chapters in this document refer to this diagram to convey the portion of the SoC pertaining to the chapter.

**Figure 2-1. What is Covered in This Chapter**



## 2.2 Features

The main architectural features are:

- Intel® Xeon® series processor Instruction Set Architecture (ISA) compatibility
- Intel® Virtualization Technology (Intel® VT-x, Intel® VT-d):
  - Intel® VT-x for Intel® IA-32 Architecture
  - Intel® VT and Intel® VT-d for Directed I/O
- Each core supports 1 hardware thread
- Each core has an L1 cache consisting of:
  - 24-KB data, parity protected
  - 32-KB instruction, parity protected
- L2 (second-level) instruction/data cache, ECC protected:
  - SoC with 2, 4 or 8 cores have 2-MB L2 per core
  - SoC with 12 or 16 cores share a 2-MB L2 per 2 cores
- 3-wide Out-of-Order (OOO) scheduler
- Improved instruction fetch and decode functions
- Improved branch prediction
- Improvements to TLB and caching hierarchy
- Hybrid OOO scheduling and OOO cache miss processing
- Per core power gating support
- Intel® Streaming SIMD Extensions 4.1 and 4.2 (SSE4.1 and SSE4.2)
- Intel® 64 bit architecture
- Support for IA-32 instruction set
- Support for Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)
- Secure Hash Algorithm New Instruction (SHA-NI)
- Support for rd_rand and rd_seed instructions
- Intel® Turbo Boost Technology (SKU dependent)
- Supported CPU Power States (C-States): C0, C1, C1e, and C6.

### 2.2.1 Goldmont New Technologies and ISA Extensions

Goldmont is the codename for the next generation 64-bit Intel® Atom™ processor cores in the SoC.

- EPT - Extended Page Tables (hardware virtualization of page tables).
- Hybrid Out-of Order (OOO) scheduling and OOO cache miss processing.
- Intel® Secure Hash Algorithm Extensions - instruction extensions that target the acceleration of the Secure Hash Algorithm (SHA).
- DRNG - Digital Random Number Generator Instructions:
  - RDRAND
  - RDSEED
- SMM Range Register - Full SMM range protection.
- Energy Efficient Turbo - Improved performance headroom.
- SMM external call trap execute disable - Security feature detects SMM code accesses outside SMRR and causes a machine check.
- APIC Virtualization; x2 APIC - Better APIC handling in Intel® VT mode.
- Pause Loop Exiting - Better spin-lock handling in Intel® VT mode.
- VMCS shadowing - Better performance of VMREAD,VMWRTE under nested virtual machine scenarios.
- CPUID Faulting - Better CPUID handling in VT mode.
- RTIT 2.0 - Improvements to make RTIT architectural.
- Memory Protection Extensions (MPX) + XSAVE - Security technology to detect buffer overflow attacks.
- "Light" Machine Check Abort (MCA) - Corrected Machine Check error Interrupt (CMCI), additional parity.
- Cache QOS - L2 cache partitioning for better real-time performance.
- Parity on IDI (Intra-Die Interconnect).
- 40-bit memory address (1,024 GB).

### 2.2.2 Intel® Virtualization Technology

Intel® Virtualization Technology (Intel® VT) makes a single system appear as multiple independent systems to the software. This allows multiple, independent operating systems to run simultaneously on a single system. Intel VT comprises technology components to support virtualization of platforms based on Intel® architecture microprocessors and chipsets. Intel® Virtualization Technology for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x) added hardware support in the processor to improve the virtualization performance and robustness. Intel® VT-x specifications and functional descriptions are included in the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

The first revision of this technology (Intel® VT-x) added hardware support in the processor to improve the virtualization performance and robustness. The latest revisions of this specification (Intel® VT-d), now called "Intel Virtualization Technology for Directed I/O, v1.3" adds direct chipset hardware hooks to improve I/O performance and robustness through DMA remapping. The SoC incorporates the Input/Output Memory Management Unit (IOMMU) (Intel® VT-d) block inside the System Agent to facilitate the translation.

### 2.2.3 Intel® VT-x Objectives

- Intel® Xeon® series processor Instruction Set Architecture (ISA) compatibility
- Robust - Virtual Machine Monitors (VMMs) no longer need to use paravirtualization or binary translation. This means that they run off-the-shelf operating systems and applications without any special steps.
- Enhanced - Intel® VT enables VMMs to run 64-bit guest operating systems on IA x86 processors.
- More Reliable - Due to the hardware support, VMMs are now smaller, less complex, and more efficient. This improves reliability and availability and reduces the potential for software conflicts.
- More Secure - Use of hardware transitions in the VMM strengthens the isolation of VMs and further prevents corruption of one VM from affecting others on the same system. Intel VT-x provides hardware acceleration for virtualization of IA platforms. VMMs use Intel VT-x features to provide improved reliable virtualized platform.

### 2.2.4 Intel® VT-x Features

- Extended Page Tables (EPT):
  - An EPT is hardware-assisted page table physical memory virtualization
  - EPT supports guest VM execution in unpaged protected / real address mode
  - EPT eliminates VM exits from guest OS to the VMM for shadow page-table Maintenance
- Virtual Processor IDs (VPID):
  - The ability to use an assigned VM Virtual Processor ID to tag processor core
- Hardware structures (such as TLBs) allowing a logic processor to cache information (such as TLBs) for multiple-linear address spaces:
  - This avoids flushes on VM transitions to give a lower-cost VM transition time
- Overall reduction in virtualization overhead
- Guest Preemption Timer:
  - A mechanism for a VMM to preempt the execution of a guest OS VM after an amount of time specified by the VMM. The VMM sets a timer value before entering a guest.
  - The feature aids VMM developers in flexibility and Quality of Service (QoS) guarantees flexibility in guest VM scheduling and building QoS schemes.
- Descriptor-Table Exiting:
  - Descriptor-table exiting allows a VMM to protect a guest OS from internal (malicious software based) attack by preventing relocation of key system data structures like Interrupt Descriptor Table (IDT), Global Descriptor Table (GDT)
- Local Descriptor Table (LDT), and Task Segment Selector (TSS):
  - A VMM using this feature intercepts (by a VM exit) attempts to relocate these data structures and prevent them from being tampered by malicious software
- VM Functions:
  - A VM function is an operation provided by the processor that is invoked using the VMFUNC instruction from guest VM without a VM exit
  - A VM function to perform EPTP switching is supported and allows guest VM to load a new value for the EPT pointer, thereby establishing a different EPT

## 2.2.5 Security and Cryptography Technologies

### 2.2.5.1 Intel® Advanced Encryption Standard New Instructions (Intel® AES-NI)

The processor supports Advanced Encryption Standard New Instructions (Intel®AES-NI) that are a set of Single Instruction Multiple Data (SIMD) instructions that enable fast and secure data encryption and decryption based on the Advanced Encryption Standard (AES). Intel® AES-NI are valuable for a wide range of cryptographic applications, for example:

- applications that perform bulk encryption/decryption
- authentication, random number generation
- authenticated encryption

AES is broadly accepted as the standard for both government and industry applications, and is widely deployed in various protocols. Intel® AES-NI consists of six Intel® SSE instructions.

Four instructions facilitate high performance AES encryption and decryption:

- AESENC
- AESENCLAST
- AESDEC
- AESDELAST

The other two, support the AES key expansion procedure.

- AESIMC
- AESKEYGENASSIST

Together, these instructions provide a full hardware AES support, offering security, high performance, and a great deal of flexibility.

### 2.2.5.2 Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions)

The Secure Hash Algorithm (SHA) is one of the most commonly employed cryptographic algorithms. Primary usages of SHA include data integrity, message authentication, digital signatures, and data de-duplication. As the pervasive use of security solutions continues to grow, SHA can be seen in more applications now than ever. The Intel® SHA Extensions are designed to improve the performance of these compute-intensive algorithms on Intel® architecture-based processors.

The Intel® SHA Extensions are a family of seven instructions based on the Intel® Streaming SIMD Extensions (Intel® SSE) that are used together to accelerate the performance of processing SHA-1 and SHA-256 on Intel architecture-based processors.

- **SHA1RNDS4**—Perform Four Rounds of SHA1 Operation
- **SHA1NEXTE**—Calculate SHA1 State Variable E after Four Rounds
- **SHA1MSG1**—Perform an Intermediate Calculation for the Next Four SHA1 Message Dwords
- **SHA1MSG2**—Perform a Final Calculation for the Next Four SHA1 Message Dwords
- **SHA256RNDS2**—Perform Two Rounds of SHA256 Operation
- **SHA256MSG1**—Perform an Intermediate Calculation for the Next Four SHA256 Message Dwords
- **SHA256MSG2**—Perform a Final Calculation for the Next Four SHA256 Message Dwords

A processor supports Intel® SHA Extensions if CPUID.(EAX=07H, ECX=0):EBX.SHA [bit 29] = 1. The Intel® SHA Extensions require only Intel® XMM™ Technology state support on operating systems, similar to SSE2 instructions.

Details of the seven instructions are in a PDF file *Intel® Architecture Instruction Set Extensions Programming Reference* available here: https://software.intel.com/en-us/isa-extensions.

### 2.2.5.3 PCLMULQDQ Instruction

The processor supports the carry-less multiplication instruction, PCLMULQDQ. PCLMULQDQ is a Single Instruction Multiple Data (SIMD) instruction that computes:

- The 128-bit carry-less multiplication of two 64-bit operands without generating Propagating carries

Carry-less multiplication is an essential processing component of several cryptographic systems and standards. Accelerating carry-less multiplication significantly contributes to achieving high-speed secure computing and communication.

### 2.2.5.4 Digital Random Number Generator

The processor introduces a software visible digital random number generation mechanism supported by a high-quality entropy source. This capability is available to programmers through the RDRAND and RDSEED instructions. The instruction is described as RDRAND—Read Random Number in Volume 2 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

Some uses of the RDRAND instruction include cryptographic key generation as used in a variety of applications including communication, digital signatures, secure storage, etc.

## 2.2.6 Intel® Turbo Boost Technology

Intel® Turbo Boost Technology is available on some of the SoC product SKUs. Refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

Intel® Turbo Boost Technology is a feature that allows the processor to opportunistically and automatically run faster than its rated operating core and/or render clock frequency when there is sufficient power headroom, and the product is within specified temperature and current limits.

The Intel® Turbo Boost Technology feature increases performance of both multi-threaded and single-threaded workloads. The processor supports a turbo mode where the processor uses the thermal capacity associated with the package and core current being drawn to determine the higher core frequencies/power levels resulting in exceeding the SoCs TDP power for short durations.

For the processor to operate with maximum performance, the turbo feature needs to be properly enabled by the BIOS and the platform must have the correct implementation of SVID IMON. Since the turbo feature is configurable and dependent on many platform design limits outside of the processor control, the maximum performance cannot be guaranteed.

Turbo mode availability is independent of the number of active cores; however, the turbo mode frequency is dynamic and dependent on the instantaneous application power load, the number of active cores, user configurable settings, operating environment, and system design.

### 2.2.6.1 Intel® Turbo Boost Technology Frequency

The processor-rated frequency assumes that all execution cores are active and are at the sustained Thermal Design Power (TDP). However, under a typical operation not all cores are active or executing a high-power workload. Most applications are consuming less than the TDP at the rated frequency. Intel Turbo Boost Technology takes advantage of the available TDP headroom and active cores are able to increase their operating frequency. To determine the highest performance frequency amongst active cores, the processor takes the following into consideration to recalculate turbo frequency during runtime:

- Number of cores operating in the C0 state
- Estimated core current consumption
- Estimated package prior and present power consumption
- Package temperature

Any of these factors affect the maximum frequency for a given workload. If the power, current, or thermal limit is reached, the processor automatically reduces the frequency to stay within its TDP limit. Turbo processor frequencies are only active if the operating system is requesting the P0 state Section 34.10, "Performance States".

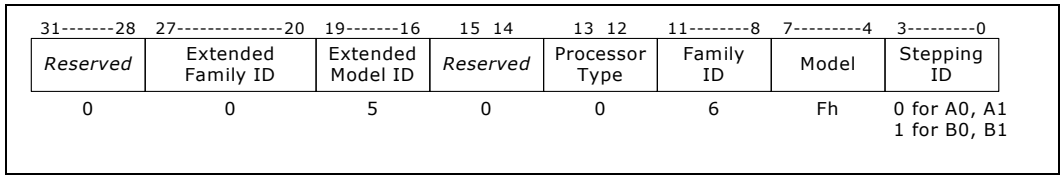
## 2.2.7 CPUID Instruction

The CPU Identification (CPUID) instruction returns processor identification and feature information to the EAX, EBX, ECX, and EDX registers, as determined by input entered in EAX (and in some cases, ECX as well).

Details of the instruction can be found in the Instruction Set Reference portion of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

When CPUID executes with EAX set to 01h, the SoC version information is returned in EAX. See Figure 2-2, "Version Information Returned by CPUID in EAX".

**Figure 2-2. Version Information Returned by CPUID in EAX**




*Note:* Once the SoC product is launched, the *Intel® 64 and IA-32 Architectures Software Developer's Manual* will refer to the SoC as having "DisplayFamily_DisplayModel" as 06_5FH.

Leaves 0 through 15h provides Basic CPUID Information.

Leaves 80000000h through 80000008h provide Extended CPUID Information.

Table 2-1, "CPUID Leaf 0h" through Table 2-21, "Extended CPUID Leaf 80000008h" contain descriptions of the leaves and sub-leaves.

### 2.2.7.1 CPUID Leaf 0 — Basic CPUID Information

**Table 2-1. CPUID Leaf 0h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 0 | n/a | EAX | [31:0] | Maximum CPUID input value in EAX that is recognized by the SoC for its Basic CPUID Information. | Normally 0x15. *See bit 22 (BOOT_NT4) of MSR 1A0h, IA32_MISC_ENABLE.* | Leaf 15h is the final leaf for SoC Basic CPUID Features |
| | | EBX | [31:0] | Intel identification, ASCII encoded "uneG" | 0x756E6547 | "Genu" |
| | | ECX | [31:0] | Intel identification, ASCII encoded "letn" | 0x6C65746E | "ntel" |
| | | EDX | [31:0] | Intel identification, ASCII encoded "Ieni" | 0x49656E69 | "ineI" |

### 2.2.7.2 CPUID Leaf 1 — Version and Feature Information

**Table 2-2. CPUID Leaf 1h, Output EAX and EBX**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 1 | n/a | EAX | [3:0] | Stepping ID | Varies by product release | A0 Stepping = 0<br>A1 Stepping = 0<br>B0 Stepping = 1 |
| | | | [7:4] | Model | 0xF | 0xF |
| | | | [11:8] | Family ID | 6 | 6 |
| | | | [13:12] | Processor Type | 0 | Original OEM Processor |
| | | | [15:14] | (Reserved) | 0 | |
| | | | [19:16] | Extended Model ID | 5 | 5 |
| | | | [27:20] | Extended Family ID | 0 | |
| | | EBX | [7:0] | Brand Index | 0 | Brand String Method used rather than Brand Index Method |
| | | | [15:8] | CLFLUSH line size | 8 | 64 bytes |
| | | | [23:16] | Maximum number of addressable IDs for logical processors in this package | 0x20 | 32 IDs Max |
| | | | [31:24] | APIC ID assigned during power up | Varies | |

## Table 2-3. CPUID Leaf 1h, Output ECX (Sheet 1 of 2)

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 1 | n/a | ECX | 0 | **SSE3** - Intel® Streaming SIMD Extensions 3 (Intel® SSE3) | 1 | Supported |
| | | | 1 | **PCLMULDQ** - PCLMULDQ instruction | 1 | Supported |
| | | | 2 | **DTES64** - 64-bit DS Area | 1 | Supported |
| | | | 3 | **MONITOR/MWAIT** feature supported | Varies | Depends on value of bit 18 in IA_MISC_ENABLE (MSR 1A0h) |
| | | | 4 | **DS_CPL** - CPL Qualified Debug Store | 1 | Supported |
| | | | 5 | **VMX** - Virtual Machine Extensions | 1 | Supported |
| | | | 6 | **SMX** - Safer Mode Extensions | 0 | Not Supported |
| | | | 7 | **EIST** - Intel Enhanced SpeedStep® Technology | 1 | Supported |
| | | | 8 | **TM2** - Intel® Thermal Monitor 2 | 1 | Supported |
| | | | 9 | **SSSE3** - Supplemental Streaming SIMD Extensions 3 | 1 | Supported |
| | | | 10 | **CNXT-ID** - L1 Context ID | 0 | Not Supported |
| | | | 11 | **Privacy MSR** | 1 | Supported |
| | | | 12 | **FMA** - **Intel®** Fast Memory Access (**Intel®** FMA) extensions using YMM state | 0 | Not Supported |
| | | | 13 | **CMPXCHG16B/CX16** - Compare and Exchange Bytes features | 1 | Supported |
| | | | 14 | **xTPR Update Control** - Allow changing bit 23 of IA32_MISC_ENABLE (MSR 1A0h) | 1 | Supported |
| | | | 15 | **PDCM** - PerfMon and Debug Capability, altering IA32_PERF_CAPABILITIES (MSR 345h) | 1 | Supported |
| | | | 16 | (Reserved) | 0 | |
| | | | 17 | **PCID** - Process-Context Identifiers | 0 | Not Supported |
| | | | 18 | **DCA** - Ability to prefetch data from memory-mapped device | 0 | Not Supported |
| | | | 19 | **SSE4.1** - Intel® Streaming SIMD Extensions 4.1 | 1 | Supported |
| | | | 20 | **SSE4.2** - Intel® Streaming SIMD Extensions 4.2 | 1 | Supported |
| | | | 21 | **x2APIC** - Extended Interrupt Mode | 1 | Supported |
| | | | 22 | **MOVBE** - MOVBE instruction | 1 | Supported |
| | | | 23 | **POPCNT** - POPCNT instruction | 1 | Supported |
| | | | 24 | **TSC-Deadline** | 1 | Supported |
| | | | 25 | **AESNI** - AESNI instruction | 1 | Supported |
| | | | 26 | **XSAVE** - XSAVE/XRSTOR extended states, XSETBV/XGETBV instructions, and XCRO | 1 | Supported |

## Table 2-3. CPUID Leaf 1h, Output ECX (Sheet 2 of 2)

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | n/a | ECX | 27 | **OSXSAVE** - A value of 1 indicates the OS has set CR4.OSXSAVE | Varies | |
| | | | 28 | **AVX** - AVX instruction extensions. | 0 | Not Supported |
| | | | 29 | **F16C/IVBNI** - 16-bit floating-point conversion instructions | 0 | Not Supported |
| | | | 30 | **RDRAND** - RDRAND instruction | 1 | Supported |
| | | | 31 | (Reserved) | 0 | |

## Table 2-4. CPUID Leaf 1h, Output EDX (Sheet 1 of 2)

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 1 | n/a | EDX | 0 | **FPU** - Floating Point Unit on Chip | 1 | Supported |
| | | | 1 | **VME** - Virtual 8086 Mode Enhancements | 1 | Supported |
| | | | 2 | **DE** - Debugging Extensions | 1 | Supported |
| | | | 3 | **PSE** - Page Size Extension | 1 | Supported |
| | | | 4 | **TSC** - Time Stamp Counter | 1 | Supported |
| | | | 5 | **MSR** - Model Specific Registers RDMSR & WRMSR Instructions | 1 | Supported |
| | | | 6 | **PAE** - Physical Address Extension | 1 | Supported |
| | | | 7 | **MCE** - Machine Check Exception | 1 | Supported |
| | | | 8 | **CX8** - CMPXCHG8B Instruction | 1 | Supported |
| | | | 9 | **APIC** - APIC On-Chip | Varies | 1 if bit 11 of IA32_APIC_BASE (MSR 1Bh) is set |
| | | | 10 | (Reserved) | 0 | |
| | | | 11 | **SEP** - SYSTENTER & SYSEXIT Instructions | 1 | Supported |
| | | | 12 | **MTRR** - Memory Type Range Registers | 1 | Supported |
| | | | 13 | **PGE** - Page Global Bit | 1 | Supported |
| | | | 14 | **MCA** - Machine Check Architecture | 1 | Supported |
| | | | 15 | **CMOV** - Conditional Move Instruction | 1 | Supported |
| | | | 16 | **PAT** - Page Attribute Table | 1 | Supported |
| | | | 17 | **PSE-36** - 36-bit Page Size Extension | 1 | Supported. Physical addresses may be up to 39 bits |
| | | | 18 | **PSN** - Processor Serial Number | 0 | Not Supported |
| | | | 19 | **CLFSH** - CLFLUSH Instruction | 1 | Supported |
| | | | 20 | (Reserved) | 0 | |
| | | | 21 | **DS** - Debug Store | 1 | Supported |
| | | | 22 | **ACPI** - Software Controlled Clock Facilities | 1 | Supported |
| | | | 23 | **MMX** - Intel MMX™ technology | 1 | Supported |

## Table 2-4. CPUID Leaf 1h, Output EDX (Sheet 2 of 2)

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | n/a | EDX | 24 | **FXSR** - FXSAVE and FXRSTOR Instructions | 1 | Supported |
| | | | 25 | **SSE** - Intel® Streaming SIMD Extensions | 1 | Supported |
| | | | 26 | **SSE2** - Intel® Streaming SIMD Extensions 2 | 1 | Supported |
| | | | 27 | **SS** - Self Snoop | 1 | Supported |
| | | | 28 | **HTT** - Max APIC IDs reserved field is Valid. See CPUID, Leaf 1, EBX[23:16] | 1 | Supported |
| | | | 29 | **TM** - Intel® Thermal Monitor | 1 | Supported |
| | | | 30 | (Reserved) | 0 | |
| | | | 31 | **PBE** - Pending Break-Enable | 1 | Supported |

### 2.2.7.3 CPUID Leaf 2 — Cache and TLB Information

**Table 2-5. CPUID Leaf 2h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 2 | n/a | EAX | [7:0] | Loop value - Number of times CPUID leaf 2 must be executed to retrieve complete information about caches and TLBs. | 0x01 | One time |
| | | | [15:8] | Micro Translation Lookaside Buffer (uTLB) Descriptor | 0xA0 | 4K pages, fully associative, 32 entries |
| | | | [23:16] | Data Translation Lookaside Buffer (DTLB) Descriptor | 0x64 | 4K pages, 4-way, 512 entries |
| | | | [30:24] | Instruction Translation Lookaside Buffer (ITLB) Descriptor | 0x61 | 4K pages, fully associative, 48 entries |
| | | | [31] | Valid bit | 0 | Bits [30:0] are valid |
| | | EBX | [7:0] | Large-Page DTLB Descriptor | 0xC2 | 2M and 4M pages, 4-way, 16 entries |
| | | | [15:8] | General | 0xFF | CPUID leaf 2 does not report cache descriptor information. Use CPUID leaf 4 instead |
| | | | [31:16] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.4 CPUID Leaf 3— Reserved

The 96-bit processor serial number is not supported. For CPUID with EAX = 3, the results in EAX, EBX, ECX and EDS are reserved and zeros are returned.

### 2.2.7.5 CPUID Leaf 4— Deterministic Cache Parameters

CPUID leaf 4 has three sub-leaves.

**Table 2-6. CPUID Leaf 4h (Sheet 1 of 2)**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 4 | 0 | EAX | [4:0] | Cache Type | 0x1 | Data Cache |
| | | | [7:5] | Cache Level | 0x1 | Level 1 (L1) |
| | | | 8 | Self Initializing Cache? | 1 | Yes |
| | | | 9 | Fully Associative? | 0 | No |
| | | | [13:10] | (Reserved) | 0 | |
| | | | [25:14] | Max number of addressable IDs for logical processors sharing this cache | 0 | Max number is 1 |
| | | | [31:26] | Max number of addressable IDs for processor cores in the SoC physical package (minus 1) | 0xF | Max number is 16 |
| | | EBX | [11:0] | System Coherency Line Size (minus 1) | 0x3F | 64 |
| | | | [21:12] | Physical Line Partitions | 0 | 1 partition |
| | | | [31:22] | Ways (minus 1) | 5 | 6 Ways |
| | | ECX | [31:0] | Sets (minus 1) | 0x3F | 64 Sets |
| | | EDX | 0 | Write-back Invalidate/Invalidate | 1 | WBINVD/INVD is not guaranteed to act upon lower level caches of non-originating threads sharing this cache. |
| | | | 1 | Cache Inclusiveness | 0 | Cache is not inclusive of lower cache levels |
| | | | 2 | Complex Cache Indexing | 1 | Complex Cache Indexing supported |
| | | | [31:3] | (Reserved) | 0 | |
| | 1 | EAX | [4:0] | Cache Type | 0x2 | Instruction Cache |
| | | | [7:5] | Cache Level | 0x1 | Level 1 (L1) |
| | | | 8 | Self Initializing Cache? | 1 | Yes |
| | | | 9 | Fully Associative? | 0 | No |
| | | | [13:10] | (Reserved) | 0 | |
| | | | [25:14] | Max number of addressable IDs for logical processors sharing this cache | 0 | Max number is 1 |
| | | | [31:26] | Max number of addressable IDs for processor cores in the SoC physical package (minus 1) | 0xF | Max number is 16 |
| | | EBX | [11:0] | System Coherency Line Size (minus 1) | 0x3F | 64 |
| | | | [21:12] | Physical Line Partitions | 0 | 1 partition |
| | | | [31:22] | Ways (minus 1) | 7 | 8 Ways |
| | | ECX | [31:0] | Sets (minus 1) | 0x3F | 64 Sets |

## Table 2-6. CPUID Leaf 4h (Sheet 2 of 2)

| | | | | | | |
|---|---|---|---|---|---|---|
| 4 | 1 | EDX | 0 | Write-back Invalidate/Invalidate | 1 | WBINVD/INVD is not guaranteed to act upon lower level caches of non-originating threads sharing this cache. |
| | | | 1 | Cache Inclusiveness | 0 | Cache is not inclusive of lower cache levels |
| | | | 2 | Complex Cache Indexing | 1 | Complex Cache Indexing supported |
| | | | [31:3] | (Reserved) | 0 | |
| | 2 | EAX | [4:0] | Cache Type | 0x3 | Unified Cache |
| | | | [7:5] | Cache Level | 0x2 | Level 2 (L2) |
| | | | 8 | Self Initializing Cache? | 1 | Yes |
| | | | 9 | Fully Associative? | 0 | No |
| | | | [13:10] | (Reserved) | 0 | |
| | | | [25:14] | Maximum number of addressable IDs for logical processors sharing this cache (minus 1) | 0x3 | Max number is 4 |
| | | | [31:26] | Maximum number of addressable IDs for processor cores in the SoC physical package (minus 1) | 0xF | Max number is 16 |
| | | EBX | [11:0] | System Coherency Line Size (minus 1) | 0x3F | 64 |
| | | | [21:12] | Physical Line Partitions | 0 | 1 partition |
| | | | [31:22] | Ways | 0xF | 16 Ways |
| | | ECX | [31:0] | Sets (minus 1) | 0x7FF | 2048 Sets |
| | | EDX | 0 | Write-back Invalidate/Invalidate | 1 | WBINVD/INVD is not guaranteed to act upon lower level caches of non-originating threads sharing this cache. |
| | | | 1 | Cache Inclusiveness | 0 | Cache is not inclusive of lower cache levels |
| | | | 2 | Complex Cache Indexing | 0 | Direct mapped cache |
| | | | [31:3] | (Reserved) | 0 | |
| | 3+ | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.6 CPUID Leaf 5— MONITOR/MWAIT

Except for EDX[31:0], values shown in Table 2-7, "CPUID Leaf 5h" are valid only when MONITOR/MWAIT instructions are enabled.

**Table 2-7. CPUID Leaf 5h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 5 | n/a | EAX | [15:0] | Smallest monitor-line size in bytes | 0x40 this can vary depending on filter size | 64 bytes or the filter size, whichever is smaller |
| | | | [31:16] | (Reserved) | 0 | |
| | | EBX | [15:0] | Largest monitor-line size in bytes | 0x40 this can vary depending on filter size | 64 bytes or the filter size, whichever is smaller |
| | | | [31:16] | (Reserved) | 0 | |
| | | ECX | 0 | Enumeration of MONITOR-WAIT extensions | 1 | Supported |
| | | | 1 | Supports treating interrupts as break-event for MWAIT, even when interrupts disabled | 1 | Supported |
| | | | [31:2] | (Reserved) | 0 | |
| | | EDX | [3:0] | Number of MWAIT C0 sub-states supported | 0 | 0 |
| | | | [7:4] | Number of MWAIT C1 sub-states supported | 2 | 2 |
| | | | [11:8] | Number of MWAIT C2 sub-states supported | 0 | 0 |
| | | | [15:12] | Number of MWAIT C3 sub-states supported | 2 | 2 |
| | | | [19:16] | Number of MWAIT C4 sub-states supported | 4 | 4 |
| | | | [23:20] | Number of MWAIT C5 sub-states supported | 2 | 2 |
| | | | [27:24] | Number of MWAIT C6 sub-states supported | 0 | 0 |
| | | | [31:28] | Number of MWAIT C7 sub-states supported | 0 | 0 |

### 2.2.7.7 CPUID Leaf 6— Digital Thermometer and Power Management

**Table 2-8. CPUID Leaf 6h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 6 | n/a | EAX | 0 | Digital Thermal Sensor (DTS) support | 1 | Supported |
| | | | 1 | Intel® Turbo Boost Technology | 0 or 1 | Support varies by product SKU. BIOS can also disable this technology |
| | | | 2 | **ARAT -** APIC-Timer-always-running feature | 1 | Supported |
| | | | 3 | (Reserved) | 0 | |
| | | | 4 | **PLN** - Power limit notification controls support | 0 | Not Supported |
| | | | 5 | **ECMD** - Clock modulation duty cycle extension support | 0 | Not Supported |
| | | | 6 | **PTM** - Package thermal management support | 0 | Not Supported |
| | | | [31:7] | (Reserved) | 0 | |
| | | EBX | [3:0] | Number of Interrupt Thresholds in Digital Thermal Sensor (DTS) | 2 | 2 thresholds |
| | | | [31:4] | (Reserved) | 0 | |
| | | ECX | 0 | Hardware Coordination Feedback Capability | 1 | Supported. IA32_MPERF (MSR E7h) and IA32_APERF (MSR E8h) are present |
| | | | [2:1] | (Reserved) | 0 | |
| | | | 3 | Performance-Energy BIAS Preference support | 0 | Not Supported IA32_ENERGY_PERFORMANCE_BIAS (MSR 1B0h) is not available |
| | | | [31:4] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.8 CPUID Leaf 7— Extended feature Flags

The output shown in Table 2-9, "CPUID Leaf 7h" depends on the ECX input value.

**Table 2-9. CPUID Leaf 7h (Sheet 1 of 2)**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 7 | 0 | EAX | [31:0] | Maximum supported sub-leaf for CPUID leaf 7 | 0 | Only sub-leaf 0 |
| | | EBX | 0 | **FSGSBASE** - Support for RDFSBASE/RDGSBASE/WRFSBASE /WRGSBASE | 1 | Supported |
| | | | 1 | Support for IA32_TSC_ADJUST (MSR 3Bh) | 1 | Supported |
| | | | 2 | **SGX** - Support for Secure Enclaves (SE) and Intel® Software Guard Extensions (Intel® SGX) | 0 | Not Supported |
| | | | 3 | **BMI1** - Bit Manipulation Set 1 | 0 | Not Supported |
| | | | 4 | **HLE** | 0 | Not Supported |
| | | | 5 | **AVX2** - Intel® Advanced Vector Extensions 2 (Intel® AVX2) | 0 | Not Supported |
| | | | 6 | (Reserved) | 0 | |
| | | | 7 | **SMEP** - Supervisor-Mode Execution Prevention | 1 | Supported |
| | | | 8 | **BMI2** - Bit Manipulation Set 2 | 0 | Not Supported |
| | | | 9 | **ERMS** - Support for Enhanced REP MOV/STOSB | 1 | Supported |
| | | | 10 | **INVPCID** - Support for INVPCID instruction | 0 | Not Supported |
| | | | 11 | **RTM** - HLE+/RTM | 0 | Not Supported |
| | | | 12 | Support for Platform Quality of Service Monitoring (PQM) capability | 0 | Not Supported |
| | | | 13 | FPU CS and FPU DS Deprecation | 1 | The processor depreciates FPU CS and FPU DS values and these fields are 0000h |
| | | | 14 | Intel® Memory Protection Extensions (Intel® MPX) | 1 | Supported |
| | | | 15 | Support for Platform Quality of Service Enforcement (PQE) | 1 | Supported |
| | | | 16 | **AVX512F** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) Foundation instructions) | 0 | Not Supported |
| | | | 17 | **AVX512DQ** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 18 | **RDSEED** - Read Random SEED instruction | 1 | Supported |
| | | | 19 | (Reserved) | 0 | |
| | | | 20 | **SMAP** - Supervisory Mode Access Protection and the LAC/STAC instructions | 1 | Supported |

## Table 2-9. CPUID Leaf 7h (Sheet 2 of 2)

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 0 | EBX | 21 | **AVX512IFMA** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 22 | **PCOMMIT** - Persistent Commit instruction | 0 | Not Supported |
| | | | 23 | **CLFLUSHOPT** - Flush a Cache Line Optimized instruction | 1 | Supported |
| | | | 24 | (Reserved) | 0 | |
| | | | 25 | Intel® Processor Trace (Intel® PT) | 1 | Supported |
| | | | 26 | **AVX512PF** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 27 | **AVX512ER** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 28 | **AVX512CD** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 29 | **SHA** - Intel® Secure Hash Algorithm Extensions (Intel® SHA Extensions) | 1 | Supported |
| | | | 30 | **AVX512BW** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | 31 | **AVX512VL** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | ECX | 0 | **PREFETCHWT1** - PREFETCHWT1 instruction | 0 | Not Supported |
| | | | 1 | **AVX512VBMI** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Not Supported |
| | | | [31:2] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| 7 | 1+ | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.9 CPUID Leaf 8 — Reserved

CPUID leaf 8 is not supported and is reserved. Zeros are returned.

### 2.2.7.10 CPUID Leaf 9 — Direct Cache Access (DCA) Information

CPUID leaf 9 is not supported and is reserved. Zeros are returned.

### 2.2.7.11 CPUID Leaf Ah — Architectural Performance Monitoring

**Table 2-10. CPUID Leaf Ah**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| A | n/a | EAX | [7:0] | Version ID of architectural performance monitoring | 4 | Version 4 |
| | | | [15:8] | Number of general-purpose, performance monitoring counters per logical processor | 4 | 4 counters |
| | | | [23:16] | Bit width of general-purpose, performance monitoring counter | 0x30 | 48 bits |
| | | | [31:24] | Length of EBX bit vector to enumerate architectural events | 7 | 7 |
| | | EBX | 0 | Core Cycle event unavailable | 0 | Core Cycle event is available |
| | | | 1 | Instruction Retired event unavailable | 0 | Instruction Retired event is available |
| | | | 2 | Reference Cycles event unavailable | 0 | Reference Cycles event is available |
| | | | 3 | Last-level cache (L2 Cache) reference event unavailable | 0 | Last-level cache (L2 Cache) reference event is available |
| | | | 4 | Last-level cache (L2 Cache) misses event unavailable | 0 | Last-level cache (L2 Cache) misses event is available |
| | | | 5 | Branch instruction retired event unavailable | 0 | Branch instruction retired event is available |
| | | | 6 | Branch mispredict retired event unavailable | 0 | Branch mispredict retired event is available |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [4:0] | Number of fixed-function performance counters (if Version ID > 1) | 3 | 3 counters |
| | | | [12:5] | Bit width of fixed-function performance counters (if Version ID > 1) | 0x30 | 48 bits |
| | | | [31:13] | (Reserved) | 0 | |

### 2.2.7.12 CPUID Leaf Bh — Extended Topology Enumeration

CPUID leaf Bh has two sub-leaves.

**Table 2-11. CPUID Leaf Bh**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| B | 0 | EAX | [4:0] | Number of bits to shift right on x2APIC ID (EDX[31:0]) to get a unique topology ID of the next level type | 1 | Shift right 1 bit |
| | | | [31:5] | (Reserved) | 0 | |
| | | EBX | [15:0] | Number of logical processors at this level type. The number reflects configuration as shipped by Intel | 1 | 1 logical processor |
| | | | [31:16] | (Reserved) | 0 | |
| | | ECX | [7:0] | Level Number - Same value as ECX input | 0 | Level 0 |
| | | | [15:8] | Level Type | 1 | SMT level |
| | | | [31:16] | (Reserved) | 0 | |
| | | EDX | [31:0] | x2APIC ID the current logical processor | Varies | |
| | 1 | EAX | [4:0] | Number of bits to shift right on x2APIC ID (EDX[31:0]) to get a unique topology ID of the next level type | 5 | Shift right 5 bits |
| | | | [31:5] | (Reserved) | 0 | |
| | | EBX | [15:0] | Number of logical processors at this level type. The number reflects configuration as shipped by Intel | 2, 4, 8, 0xB, or 0x10 | 2, 4, 8, 12, or 16 depending on product SKU |
| | | | [31:16] | (Reserved) | 0 | |
| | | ECX | [7:0] | Level Number - Same value as ECX input | 1 | Level 1 |
| | | | [15:8] | Level Type | 2 | Core level |
| | | | [31:16] | (Reserved) | - | 0 |
| | | EDX | [31:0] | x2APIC ID the current logical processor | Varies | |
| | 2+ | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | Original ECX | 2+ | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.13 CPUID Leaf Ch — Reserved

CPUID leaf Ch is not supported and is reserved. Zeros are returned.

### 2.2.7.14 CPUID Leaf Dh — Intel® Memory Protection Extensions (Intel® MPX), XSAVE Feature

CPUID leaf Dh has six sub-leaves (0 through 4, and 8) and is shown in Table 2-12, "CPUID Leaf Dh". The other sub-leaves are reserved.

## Table 2-12. CPUID Leaf Dh (Sheet 1 of 2)

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| D | 0 | EAX | 0 | **X87 State** - Legacy Floating Point (x87/MMX) | 1 | Legacy FP state can be managed by XSAVE |
| | | | 1 | **SSE State** - Intel® Streaming SIMD Extensions (Intel® SSE) | 1 | Intel® SSE state can be managed by XSAVE |
| | | | 2 | **AVX State** - Advanced Vector Extensions (Intel® AVX) | 0 | Intel® AVX state cannot be managed by XSAVE |
| | | | 3 | **BNDREGS State** - Intel® Memory Protection Extensions (Intel® MPX) BNDREGS | 1 | Intel® MPX Bound Registers (BNDREGS) can be managed by XSAVE |
| | | | 4 | **BNDCSR State** - Intel® Memory Protection Extensions (Intel® MPX) BNDCSR | 1 | Intel® MPX Bound Control and Status Register (BNDCSR) can be managed by XSAVE |
| | | | [7:5] | **AVX-512 State** - Intel® Advanced Vector Extensions 512 (Intel® AVX-512) | 0 | Intel® AVX-512 not supported and cannot can be managed by XSAVE |
| | | | 8 | **PT State** - Intel® Processor Trace (Intel® PT), used for IA32_XSS (MSR DA0h) | 0 | Intel® PT MSRs state cannot can be managed by XSAVE |
| | | | 9 | **PKRU State** - Protection-key feature's register PKRU | 0 | PKRU state cannot can be managed by XSAVE |
| | | | [31:10] | (Reserved) | 0 | |
| | | EBX | [31:0] | Maximum size required for features enabled in XCR0 | 0x240 | 576 bytes |
| | | ECX | [31:0] | Max size required by all processor supported features | 0x440 | 1088 bytes |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 1 | EAX | 0 | Availability of XSAVEOPT | 1 | Available |
| | | | 1 | Support of XSAVEC and compact extensions of legacy XSTROR | 1 | Supported |
| | | | 2 | Support of XGETBV Leaf 1 | 1 | Supported |
| | | | 3 | Support of XSAVES and XRSTORS instructions, and IA32_XSS (MSR DA0h) | 1 | Supported |
| | | | [31:4] | (Reserved) | 0 | |
| | | EBX | [31:0] | Maximum size required for features enabled in XCR0 \| IA32_XSS (MSR DA0h) | 0x240 | 576 bytes |
| | | ECX | [7:0] | (Reserved) | 0 | |
| | | | 8 | **PT State** - Intel® Processor Trace (Intel® PT) | 1 | Corresponding bit in IA32_XSS (MSR DA0h) can be set |
| | | | [31:9] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

## Table 2-12. CPUID Leaf Dh (Sheet 2 of 2)

| | | | | | | |
|---|---|---|---|---|---|---|
| D | 2 | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 3 | EAX | [31:0] | Intel® Memory Protection Extensions (Intel® MPX) - Size of feature state-save area | 0x40 | 64 bytes |
| | | EBX | [31:0] | Intel® MPX - Offset of feature state-save area | 0x3C0 | 960 bytes |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 4 | EAX | [31:0] | Intel® MPX - Size of feature state-save area | 0x40 | 64 bytes |
| | | EBX | [31:0] | Intel® MPX - Offset of feature state-save area | 0x400 | 1024 bytes |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 8 | EAX | [31:0] | Intel® Processor Trace (Intel® PT) - Size of Intel® PT state-save area | 0x80 | 128 bytes |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | 0 | Intel® PT - Supervisor State | 1 | Intel® PT is supported in IA32_XSS (MSR DA0h) |
| | | | [31:1] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | All other values of ECX | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.15 CPUID Leaf Eh — Reserved

CPUID leaf Eh is not supported and is reserved. Zeros are returned.

### 2.2.7.16 CPUID Leaf Fh — Platform and Cache Quality of Service (QoS)

CPUID leaf Fh, Platform QoS Monitoring Enumeration and Cache QoS Monitoring Capability, is not supported and is reserved. Zeros are returned.

### 2.2.7.17 CPUID Leaf 10h — Platform and Cache QoS Enforcement Enumeration

CPUID leaf 10h contains three sub-leaves and is show in Table 2-13, "CPUID Leaf 10h".

**Table 2-13. CPUID Leaf 10h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 10 | 0 | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | 0 | (Reserved) | 0 | |
| | | | 1 | L3 Cache | 0 | SoC has no L3 Cache |
| | | | 2 | L2 Cache | 1 | Supports L2 Cache QoS Enforcement |
| | | | [31:3] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 1 | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 2 | EAX | [4:0] | Length of Masks (minus 1) | 7 | 8 |
| | | | [31:5] | (Reserved) | 0 | |
| | | EBX | [31:0] | Bitmask | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [15:0] | Highest COS number supported for this ResID (minus 1) | 3 | 4 is highest number |
| | | | [31:16] | (Reserved) | 0 | |
| | 3+ | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.18 CPUID Leaf 11h — Reserved

CPUID leaf 11h is not supported and is reserved. Zeros are returned.

### 2.2.7.19 CPUID Leaf 12h — Reserved

CPUID leaf 12h is not supported and is reserved. Zeros are returned.

### 2.2.7.20 CPUID Leaf 13h — Reserved

CPUID leaf 13h is not supported and is reserved. Zeros are returned.

### 2.2.7.21 CPUID Leaf 14h — Intel® Processor Trace (Intel® PT) Enumeration

CPUID leaf 14h has two sub-leaves and is shown in Table 2-14, "CPUID Leaf 14h".

**Table 2-14. CPUID Leaf 14h (Sheet 1 of 2)**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 14 | 0 | EAX | [31:0] | Maximum valid sub-leaf | 1 | Only sub-leaves 0 and 1 are valid |
| | | EBX | 0 | CR3 Filtering Support | 1 | IA32_RTIT_CTL (MSR 570h) can be set to 1, and that IA32_RTIT_CR3_MATCH (MSR 572h) can be accessed |
| | | | 1 | Support for Cycle-Accurate Mode and Configurable PSB | 1 | Supported |
| | | | 2 | Support for IP filtering, TraceStop filtering, and MTC timing packet | 1 | Supported |
| | | | 3 | Support for Processor Trace MSRs preserved across warm reset | 1 | Supported |
| | | | [31:4] | (Reserved) | 0 | |
| | | ECX | 0 | Support of ToPA Output scheme | 1 | Supported. Tracing can be enabled with the ToPA bit of IA32_RTIT_CTL (MSR 570h) |
| | | | 1 | ToPA Tables Support Multiple Output Regions | 1 | Supported |
| | | | 2 | Support of Single-Range Output scheme | 1 | Supported |
| | | | 3 | Support of output to Trace Transport subsystem | 0 | Not Supported |
| | | | [30:2] | (Reserved) | 0 | |
| | | | 31 | Support of IP payloads contain LIP | 1 | Generated packets which contain IP payloads contain LIP values, which include the CS base component |
| | | EDX | [31:0] | (Reserved) | 0 | |
| | 1 | EAX | [1:0] | Number of configurable Address Ranges for filtering supported | 2 | 2 ranges |
| | | | [15:2] | (Reserved) | 0 | |
| | | | [31:16] | Bitmap of supported MTC Period encodings | 0x249 | 0x249 |
| | | EBX | [15:0] | Bitmap of supported Cycle Threshold Value encodings | 0xFFFF | 0xFFFF |
| | | | [31:16] | Bitmap of supported configurable PSB Frequency encodings | 0x003F | 0x003F |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

**Table 2-14. CPUID Leaf 14h (Sheet 2 of 2)**

| | | | | | | |
|---|---|---|---|---|---|---|
| 14 | 2+ | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.22 CPUID Leaf 15h — Time Stamp Counter (TSC) and Crystal Clock Ratio

CPUID leaf 15h is the last Basic CPUID leaf.

**Table 2-15. Basic CPUID Leaf 15h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 15 | n/a | EAX | [31:0] | Denominator of TSC/crystal clock ratio | 3 | 3 |
| | | EBX | [31:0] | Numerator of TSC/crystal clock ratio | 0xC0 | 192 |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.23 CPUID Leaves 16h through 7FFFFFFFh — Reserved

CPUID leaves 16h through 7FFFFFFFh are not supported and are reserved. Each of these leaves produces the same EAX, EBX, ECX, and EDS values as leaf 15h.

CPUID leaves greater than 80000000h are also reserved and produce the same EAX, EBX, ECX, and EDS values as leaf 15h.

### 2.2.7.24 CPUID Leaf 80000000h — Maximum EAX Value for CPUID Instruction

**Table 2-16. Extended CPUID Leaf 80000000h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000000 | n/a | EAX | [31:0] | Maximum input value in EAX for Extended Function CPUID information | 0x80000008 | 0x80000008 is the maximum value |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.25 CPUID Leaf 80000001h — Extended Feature Flags

**Table 2-17. Extended CPUID Leaf 80000001h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000001 | n/a | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | 0 | Availability of LAHF and SAHF instructions in 64-bit mode | 1 | Available |
| | | | 5 | Availability of LZCNT instruction | 0 | Not Available |
| | | | [7:1] | (Reserved) | 0 | |
| | | | 8 | Availability of PREFETCHW instruction | 1 | Available |
| | | | [31:9] | (Reserved) | 0 | |
| | | EDX | [10:0] | (Reserved) | 0 | |
| | | | 11 | Availability of SYSCALL and SYSTRET instructions in 64-bit mode | 1 | Available |
| | | | [19:12] | (Reserved) | 0 | |
| | | | 20 | Availability of the Execute Disable Bit | Varies | 1 indicates Available. Will be 1 whenever the NX_DISABLE bit of IA32_CR_MISC_ENABLES (MSR 1A0h) = 0 |
| | | | [25:21] | (Reserved) | 0 | |
| | | | 26 | 1GB Pages | 1 | Enabled |
| | | | 27 | RDTSCP/IA32_TSC_AUX | 1 | 1 |
| | | | 28 | (Reserved) | 0 | |
| | | | 29 | Availability of Intel® 64 Architecture | 1 | Available |
| | | | [31:30] | (Reserved) | 0 | |

### 2.2.7.26 CPUID Leaves 80000002h, 80000003h, and 80000004h— Intel Processor Brand String

**Table 2-18. Extended CPUID Leaves 80000002h, 80000003h, and 80000004h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000002 | n/a | EAX | [31:0] | Processor Brand String | 65746E49 | |
| | | EBX | [31:0] | | 2952286C | |
| | | ECX | [31:0] | | 6F744120 | |
| | | EDX | [31:0] | | 4D54286D | |
| 80000003 | n/a | EAX | [31:0] | | 50432029 | |
| | | EBX | [31:0] | | 33432055 | |
| | | ECX | [31:0] | | 20383538 | |
| | | EDX | [31:0] | | 2E322040 | |
| 80000004 | n/a | EAX | [31:0] | | 48473030 | |
| | | EBX | [31:0] | | 7A | |
| | | ECX | [31:0] | | 0 | |
| | | EDX | [31:0] | | 0 | |

### 2.2.7.27 CPUID Leaf 80000005h — Reserved

CPUID leaf 80000005h is not supported and is reserved. Zeros are returned.

### 2.2.7.28 CPUID Leaf 80000006h — Cache Parameters

**Table 2-19. Extended CPUID Leaf 80000006h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000006 | n/a | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [7:0] | Cache Line Size | 0x40 | 64 |
| | | | [15:12] | Associativity field | 0x8 | 8 |
| | | | [31:16] | Cache size in 1k units | 0x800 | 2048 1k units (2MB) |
| | | EDX | [31:0] | (Reserved) | 0 | |

### 2.2.7.29 CPUID Leaf 80000007h — Advanced Power Management

**Table 2-20. Extended CPUID Leaf 80000007h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000007 | n/a | EAX | [31:0] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [7:0] | (Reserved) | 0 | |
| | | | 8 | Always Running TSC | 1 | Available |
| | | | [31:9] | (Reserved) | 0 | |

### 2.2.7.30 CPUID Leaf 80000008h — Virtual/Physical Address Sizes

**Table 2-21. Extended CPUID Leaf 80000008h**

| Leaf [EAX] (Hex) | Sub-leaf [ECX] (Hex) | CPUID Results | Bits | Meaning | Returned Value | What Value Indicates |
|---|---|---|---|---|---|---|
| 80000008 | n/a | EAX | [7:0] | Number of Physical Address Bits | 0x27 | 39 bits |
| | | | [15:8] | Number of Linear Address Bits | 0x30 | 48 bits |
| | | | [31:16] | (Reserved) | 0 | |
| | | EBX | [31:0] | (Reserved) | 0 | |
| | | ECX | [31:0] | (Reserved) | 0 | |
| | | EDX | [31:0] | (Reserved) | 0 | |

§ §

# 3 System Address Map

In addition to the Intel Atom® cores, the SoC contains two other units, the Innovation Engine (IE) and Intel® Management Engine (ME), that execute the Intel® architecture instruction set. Each has its own Memory Space, I/O Space and PCI domain. This document refers to these three units as:

- Host Root Space - Associated with the Intel Atom® cores
- IE Root Space - Associated with the IE instruction-execution unit
- Intel® ME Root Space - Associated with the Intel® ME instruction-execution unit

This Chapter describes the Host Root Space, which includes the Configuration Space, Memory Space, and I/O Space of the SoC. Information is provided to show Memory Address Space configuration and how to access Sideband Registers and Configuration Space.

Model-Specific Registers (MSRs) are accessed by executing Read MSR (RDMSR) and Write MSR (WRMSR) instructions. CPU Identification Registers are accessed via CPUID instruction.

## 3.1 Host Root Space—Memory Space

The Memory Space address map for the Host Root Space is shown in Table 3-1, "Host Memory Space Address Map". The Memory-Mapped I/O (MMIO) that is relocatable in Host Memory Space is shown in Table 3-2, "Relocatable MMIO in Host Memory Space".

The SoC implements 40 address bits providing 1,024 GB of addressable Host Root Memory Space for use by the CPU and devices. This is the address space accessible by memory reads and writes. These addresses are generated by the processor and also by integrated and external I/O devices. The target of the read and write transactions can be system memory or integrated memory and external I/O devices. For PCI compliance the devices and SoC internal circuitry provide a full 64-bit address space. Even so, addresses greater than 1,024 GB (40 address bits) never address a physical resource in the SoC, and are always terminated with an error.

*Note:* The ultimate subtractive-decode agent in Host Root Space is the Primary-to-Sideband Bridge (P2SB) and not the Low Pin Count (LPC) controller. Any transaction reaching the P2SB which does not match a valid mapped region is terminated. Such transactions are either dropped for Posted Requests or result in an Unsupported Request (UR) for Non-Posted Requests.

### 3.1.1 Host Memory Space—Regions and Boundaries

**Table 3-1. Host Memory Space Address Map (Sheet 1 of 3)**

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| **DOS DRAM (0 to 1 MB)** | | | |
| DRAM<br>[640 KB] | 0000_0000 | 0009_FFFF | DOS_DRAM |
| VGA<br>(A Segment and B Segment)<br>[128 KB] | 000A_0000 | 000B_FFFF | A and B segments' VGA region of memory must be configured before first access and may target one and only one of these:<br>• PCIe Root Port which has VGAE (bit 3) of its Bridge Control Register (BCTL) is set.<br>• Legacy Block<br>• DRAM in SBFT mode when ABSEGINDRAM (bit 2) of the B_CR_BMISC_0_0_0_MCHBAR register is set.<br><br>***Note:*** The SoC does not support SMM in the A and B segments of memory (the VGA region). |
| DRAM<br>(C Segment and D Segment)<br>[128 KB] | 000C_0000 | 000D_FFFF | The SoC does not support ISA Expansion region. This area always maps to system DRAM.<br>Access is from the CPU only, no inbound access support—an inbound access will be aborted. |
| Extended System BIOS<br>(E Segment)<br>[64 KB] | 000E_0000 | 000E_FFFF | Reads and Writes to E segment are to DRAM.<br><br>***Note:*** The SoC does not support reading from the BIOS boot interface via the E segment.<br><br>Access from CPU only, no inbound access support. |

## Table 3-1. Host Memory Space Address Map (Sheet 2 of 3)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| System BIOS (Upper) (F segment) [64 KB] | 000F_0000 | 000F_FFFF | Reads and Writes to F segment are to DRAM.<br><br>***Note:*** The SoC does not support reading from the BIOS boot interface via the F segment.<br><br>Access from CPU only, no inbound access support. |
| **Lower DRAM Window (1 MB to TOLUD minus 1)** | | | |
| DRAM [1 MB] | 00F0_0000 | 0100_0000 | The SoC does not support ISA hole. This range is always mapped to DRAM. |
| TSEG | B-unit System Management Mode Memory Base (TSEGMB_0_0_0_PCI) | BGSM_0_0_0_PCI minus 1 | TSEG is locked by BIOS by setting the appropriate levels in TSEGMB_0_0_0_PCI and BGSM_0_0_0_PCI. |
| BGSM | B-unit Graphics Stolen Memory Base (BGSM_0_0_0_PCI) | TOLUD minus 1 | For the SoC, the BGSM_0_0_0_PCI range is not applicable. Even so, BGSM_0_0_0_PCI must be programmed since it affects TSEG. |
| **Low MMIO (TOLUD to 4 GB)** | | | |
| ABORT | FEB0_0000 | FEBF_FFFF | CRAB Abort page region |
| IOxAPIC | FEC0_0000 | FEC0_0040 | IOxAPIC space.<br>Within FECx_x000 to FECx_x040, BIOS configures the window using APIC Range Select (IOAC.ASEL) and APIC Enable (IOAC.AEN). The IOAC register is in the P2SB Bridge device.<br><br>***Note:*** Intel only validates the range FEC0_0000 to FEC0_0040. |
| HPET | FED0_0000 | FED0_03FF | High performance event timer.<br>Although FED0_0000 through FED0_33FF are configurable, only FED0_0000 through FED0_03FF is expected. |
| TPM | FED4_0000 | FED4_0FFF | TPM1.2 range |
| xHCI DbC | FED6_0000 | FED6_0FFF | The OS must be informed that this address range is reserved. No other MMIO must overlap range. |
| Local APIC | FEE0_0000 | FEEF_FFFF | APIC space used to send MSIs to the CPU and INTR ACKs to legacy block. |
| BIOS4 - Feature space for LPC | FF00_0000 | FF0F_FFFF | BDE.0 |
| | FF10_0000 | FF1F_FFFF | BDE.1 |
| | FF20_0000 | FF2F_FFFF | BDE.2 |
| | FF30_0000 | FF3F_FFFF | BDE.3 |
| BIOS4 - Data space for SPI and LPC | FF40_0000 | FF4F_FFFF | BDE.0 |
| | FF50_0000 | FF5F_FFFF | BDE.1 |
| | FF60_0000 | FF6F_FFFF | BDE.2 |
| | FF70_0000 | FF7F_FFFF | BDE.3 |

## Table 3-1. Host Memory Space Address Map (Sheet 3 of 3)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| **Low MMIO (TOLUD to 4 GB) (continued)** | | | |
| BIOS3 - Feature space for LPC | FF80_0000 | FF87_FFFF | BDE.8 |
| | FF88_0000 | FF8F_FFFF | BDE.9 |
| | FF90_0000 | FF97_FFFF | BDE.10 |
| | FF98_0000 | FF9F_FFFF | BDE.11 |
| | FFA0_0000 | FFA7_FFFF | BDE.12 |
| | FFA8_0000 | FFAF_FFFF | BDE.13 |
| | FFB0_0000 | FFB7_FFFF | BDE.14 |
| BIOS3 - Feature space for LPC | FFB8_0000 | FFBF_FFFF | BDE.15; always enabled |
| BIOS3 - Data space for SPI and LPC | FFC0_0000 | FFC7_FFFF | BDE.8 |
| | FFC8_0000 | FFCF_FFFF | BDE.9 |
| | FFD0_0000 | FFD7_FFFF | BDE.10 |
| | FFD8_0000 | FFDF_FFFF | BDE.11 |
| | FFE0_0000 | FFE7_FFFF | BDE.12 |
| | FFE8_0000 | FFEF_FFFF | BDE.13 |
| | FFF0_0000 | FFF7_FFFF | BDE.14 |
| BIOS2 - Data space for SPI and LPC | FFF8_0000 | FFFB_FFFF | BDE.15; always enabled |
| BIOS - Data space for SPI and LPC | FFFC_0000 | FFFF_FFFF | This BIOS region is enabled by default and can not be disabled. Reset boot vector is FFFF_FFF0. |
| ECAM Base Address | PCIEXBAR_LO_0_0_0_PCI and PCIEXBAR_HI_0_0_0_PCI | PCIEXBAR plus 256 MB in 32-bit address space | PCI Express Enhanced Configuration Access Mechanism (ECAM) location. PCI and PCIe Configuration Space accessible as memory-mapped accesses. |
| Protected Memory (Low) for VT-d | PLMBASE_VTDBAR | PLMLIMIT_VTDBAR | Area of protected memory located <4GB for VT-d. |
| **4 GB to TOUUD-1 -- High DRAM Window** | | | |
| High DRAM | 1_0000_0000 | TOUUD-1 | High Main Memory. Top of Upper Usable DRAM. |
| **TOUUD to 1024 GB -- High MMIO** | | | |
| High MMIO | TOUUD | 1024 GB | High MMIO for 64-bit MMIO |
| Protected Memory (High) for VT-d | PHMBASE_VTDBAR | PHMLIMIT_VTDBAR | Area of protected memory located >4GB for VT-d. |

## 3.1.2 Host Memory Space—Relocatable Memory-Mapped I/O

Table 3-2, "Relocatable MMIO in Host Memory Space" lists the Memory-Mapped I/O that is relocatable in Host Memory Space. The I/O listed as being in 32-bit address space can be configured within the "Low MMIO" region shown in Table 3-1, "Host Memory Space Address Map". The I/O listed as being in 64-bit address space can be configured within the "High MMIO or Low MMIO" region of the same table.

**Table 3-2. Relocatable MMIO in Host Memory Space (Sheet 1 of 5)**

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| Dedicated Integrated Root Port for Intel® QuickAssist Technology | D6:F0<br>MEMBASE<br>MEMLIMIT | variable length in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D6:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| Intel® QuickAssist Technology | Bus x:D0:F0<br>PSECRAMUBAR<br>PSECRAMLBAR | 512 KB in 64-bit address space | Intel® QuickAssist Technology, PF Memory Base Address Register |
| PCI Express RP[0]<br>*Root Port Cluster 0, Root Port 0* | D9:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D9:F0<br>MEMBASE<br>MEMLIMIT | variable length in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D9:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| PCI Express RP[1]<br>*Root Port Cluster 0, Root Port 1* | D10:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D10:F0<br>MEMBASE<br>MEMLIMIT | variable length in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D10:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| PCI Express RP[2]<br>*Root Port Cluster 0, Root Port 2* | D11:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D11:F0<br>MEMBASE<br>MEMLIMIT | variable length in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D11:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |

## Table 3-2. Relocatable MMIO in Host Memory Space (Sheet 2 of 5)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| PCI Express RP[3]<br>*Root Port Cluster 0, Root Port 3* | D12:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB<br>in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D12:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D12:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| PCI Express RP[4]<br>*Root Port Cluster 1, Root Port 0* | D14:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB<br>in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D14:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D14:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| PCI Express RP[5]<br>*Root Port Cluster 1, Root Port 1* | D15:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB<br>in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D15:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D15:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| PCI Express RP[6]<br>*Root Port Cluster 1, Root Port 2* | D16:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB<br>in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D16:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D16:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |

## Table 3-2. Relocatable MMIO in Host Memory Space (Sheet 3 of 5)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| PCI Express RP[7]<br>*Root Port Cluster 1, Root Port 3* | D17:F0<br>EXPPTMBAR<br>(BAR1,BAR0) | 128 KB<br>in 64-bit address space | As defined by the Base Address Register 0 (BAR0) and Base Address Register 1 (BAR1) in the Function Configuration Space. |
| | D17:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D17:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| SMBus 2.0 Controller - Host | D18:F0<br>SMTBAR<br>(BAR1,BAR0) | 1 KB<br>in 64-bit address space | SMBus 2.0 Controller (Host) |
| SATA Controller 0 | D19:F0<br>ABAR<br>(BAR5) | 2 KB<br>in 32-bit address space | AHCI Base Address for SATA 3.1 Controller 0 |
| SATA Controller 1 | D20:F0<br>ABAR<br>(BAR5) | 2 KB<br>in 32-bit address space | AHCI Base Address for SATA 3.1 Controller 1 |
| USB Combo Controller | D21:F0<br>MBAR<br>(BAR1,BAR0) | 64 KB in 64-bit address space | USB3/USB2 combo controller |
| Dedicated Integrated Root Port for LAN Controller 0 | D22:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D22:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |
| LAN Controller 0 | Bus y:D0:Fn<br>(BAR1, BAR0) | 128 KB<br>in 64-bit address space | LAN port controller, independent per function |
| | Bus y:D0:Fn<br>BAR2 | 32B<br>in 64-bit address space | LAN port controller, independent per function |
| | Bus y:D0:Fn<br>(BAR5, BAR4) | 16 KB<br>in 64-bit address space | LAN port MSI-X, independent per function |
| Dedicated Integrated Root Port for LAN Controller 1 | D23:F0<br>MEMBASE<br>MEMLIMIT | variable length<br>in 32-bit address space | As defined by the Memory Base register and Memory Limit register in the Function Configuration Space. |
| | D23:F0<br>PFBASEU<br>PFBASE<br>and<br>PFLIMITU<br>PFLIMIT | variable length<br>in 64-bit address space | As defined by the Prefetchable Memory Base register (Upper and Lower) and Prefetchable Memory Limit register (Upper and Lower) in the Function Configuration Space. The two Upper registers are for 64-bit addressing. |

## Table 3-2. Relocatable MMIO in Host Memory Space (Sheet 4 of 5)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| LAN Controller 1 | Bus z:D0:Fn (BAR1, BAR0) | 128 KB in 64-bit address space | LAN port controller, independent per function |
| | Bus z:D0:Fn BAR2 | 32B in 64-bit address space | LAN port controller, independent per function |
| | Bus z:D0:Fn (BAR5, BAR4) | 16 KB in 64-bit address space | LAN port MSI-X, independent per function |
| Intel® ME HECI 1 | D24:F0 HECI1_MMIO_MBAR | 4 KB in 64-bit address space | |
| Intel® ME HECI 2 | D24:F1 HECI2_MMIO_MBAR | 4 KB in 64-bit address space | |
| Intel® ME KT | D24:F3 KT_HOST_MEMBAR | 4 KB in 32-bit address space | Intel® ME Keyboard and Text (KT) Redirection |
| Intel® ME HECI 3 | D24:F4 HECI3_MMIO_MBAR | 4 KB in 64-bit address space | |
| HS-UART Controller 0 | D26:F0 MEMBA | 256 Bytes in 32-bit address space | HS-UART |
| HS-UART Controller 1 | D26:F1 MEMBA | 256 Bytes in 32-bit address space | HS-UART |
| HS-UART Controller 2 | D26:F2 MEMBA | 256 Bytes in 32-bit address space | HS-UART |
| IE HECI 1 | D27:F0 HECI1_MMIO_MBAR | 4 KB in 64-bit address space | |
| IE HECI 2 | D27:F1 HECI2_MMIO_MBAR | 4 KB in 64-bit address space | |
| IE KT | D27:F3 KT_HOST_MEMBAR | 4 KB in 32-bit address space | IE Keyboard and Text (KT) Redirection |
| IE HECI 3 | D27:F4 HECI1_MMIO_MBAR | 4 KB in 64-bit address space | |
| eMMC | D28:F0 BAR_HIGH, BAR | 4 KB in 64-bit address space | |
| LPC | D31:F0 LGMR | 64 KB in 32-bit address space | LPC memory space |
| Proxy (P2SB) | D31:F1 SBREG_BAR | 16 MB in 64-bit address space | P2SB register aperture |

## Table 3-2. Relocatable MMIO in Host Memory Space (Sheet 5 of 5)

| Memory Address Range Name [Size] | Start Address or Base Address (Hexadecimal) | End Address or Range (Hexadecimal) | Dependency/Comments |
|---|---|---|---|
| PMC | D31:F2<br>PBASE | 512 B in 32-bit address space | PMC memory space |
| SMBus 2.0 Controller - Legacy | D31:F4<br>SMBMBAR | 256B in 64-bit space as configured in PSF address space | SMBus 2.0 Controller (Legacy). Used for SPD interface to memory modules. |
| SPI Controller | D31:F5<br>BIOS_SPI_BAR0 | 4KB in 32-bit address space | SPI memory space |

## 3.2 Host Root Space—I/O Space

The SoC implements 64 kilobytes plus three (65,539 decimal) bytes of I/O Space addressing. For I/O instructions that access the highest-addressed ports (FFFDh - FFFFh), there are situations where the three bytes beyond the 64-KB boundary are accessed. There are no resources for these highest-addressed three bytes.

For example, a four-byte read of I/O address FFFEh results in reading two data bytes within the 64-KB boundary and two data bytes beyond the 64-KB boundary. The latter two data bytes have no resources and must be ignored.

The location of the I/O Port register being written to, or read from, via an I/O instruction (IN, INS, OUT, OUTS) is calculated by the SoC. The register can be physically located within the SoC, LPC device, SPI device, or on a PCI Express* device connected to one of the integrated Root Ports.

The primary subtractive-decode agent in Host Root Space is the Primary-to-Sideband Bridge (P2SB) and not the Low Pin Count (LPC) controller. Any transaction reaching the P2SB which does not match a valid mapped region is terminated. Such transactions are either dropped for Posted Requests or result in an Unsupported Request (UR) for Non-Posted Requests.

***Warning:*** Each SoC PCIe* integrated Root Port can be configured by the Operating System (OS) and BIOS to claim a portion of I/O space addresses. When an I/O instruction accesses an I/O Port within this claimed portion, the transaction is routed to the devices on the particular Root Port. The OS and BIOS must ensure that the integrated PCIe Root Ports' claimed portions of I/O space are configured to not overlap the addresses of the I/O Port registers, mostly legacy device registers, located within the SoC.

The fixed-address I/O Port registers are in Table 3-3, "Host Fixed I/O Address Space".

The relocatable (BARs) I/O Port registers are in Table 3-4, "Host Relocatable I/O Address Space".

### 3.2.1 Host I/O Space—Fixed I/O Addresses

See Table 3-3, "Host Fixed I/O Address Space". The SoC calculates the physical location of an I/O Port register based on a number of conditions. The fixed-address ranges are either positively decoded in the System Agent or subtractively routed to the Primary to Sideband Bridge (P2SB).

In Table 3-3, "Host Fixed I/O Address Space":

- For address ranges listed as having NEG (negative) address decode, an I/O instruction is routed to the device listed in the table. This happens only if there is no enabled device with a BAR that instead captures the I/O address. If there is no BAR that captures the transaction, then the target device of the NEG decode must be enabled.
  - If the target device of the NEG decode is disabled, transactions to the address range are terminated.
- When an address range is listed as having a POS (positive) address decode and the address range is enabled, the I/O transaction is routed to a relocatable BAR. See Table 3-4, "Host Relocatable I/O Address Space". The target device responding to the I/O address can be integrated within or downstream of one of the PCIe Root Ports if the address is within the Root Port's I/O Base and Limit registers.
  - If the target device is disabled, transactions to the address range are terminated.
  - If the POS address range is disabled, the transaction is routed to the LPC controller.
- For address ranges listed as having Fabric address decode, the destination of the I/O transaction is determined by registers and configuration settings of the SoC. Falling into this category are COM1/COM2 I/O address which can be assigned to HSUART and the VGA Enabled (BCTL.VGAE=1) of the Root Ports. Here, if a positive decode of the I/O address is made, the I/O transaction is routed to the appropriate destination.

## Table 3-3. Host Fixed I/O Address Space (Sheet 1 of 4)

| Address (hex) | Address Decode | Targets | Register Control |
|---|---|---|---|
| 00 - 1F | NEG | Transaction is terminated | |
| 20 - 21 | NEG | 8259 PIC | |
| 24 - 25 | NEG | 8259 PIC | |
| 28 - 29 | NEG | 8259 PIC | |
| 2C - 2D | NEG | 8259 PIC | |
| 2E - 2F | NEG | LPC | |
| 30 - 31 | NEG | 8259 PIC | |
| 34 - 35 | NEG | 8259 PIC | |
| 38 - 39 | NEG | 8259 PIC | |
| 3C - 3D | NEG | 8259 PIC | |
| 40 | NEG | 8254 Timer | |
| 41 | NEG | Transaction is terminated | |
| 42 - 43 | NEG | 8254 Timer | |
| 4E - 4F | NEG | LPC | |
| 50 | NEG | 8254 Timer | |
| 51 | NEG | Transaction is terminated | |
| 52 - 53 | NEG | 8254 Timer | |
| 60 | NEG | PS/2 Legacy Keyboard/Mouse | |
| 61 | NEG | NMI Controller | |
| 62 | NEG | LPC | |
| 63 | NEG | NMI Controller | |
| 64 | NEG | PS/2 Legacy Keyboard/Mouse | |
| 65 | NEG | NMI Controller | |
| 66 | NEG | LPC | |
| 67 | NEG | NMI Controller | |
| 70 | NEG | CPU Interface, RTC, Power Management Controller | |
| 71 | NEG | RTC, Power Management Controller | |
| 72 - 73 | NEG | RTC, Power Management Controller | |
| 74 | NEG | RTC, Power Management Controller | |
| 75 | NEG | RTC, Power Management Controller | |
| 76 - 77 | NEG | RTC, Power Management Controller | |
| 80 | POS | LPC or PCIe Root Port | RCFG.RPRID = 1 |
| 81 - 83 | NEG | Transaction is terminated | |
| 84 - 86 | POS | LPC or PCIe Root Port | RCFG.RPRID = 1 |
| 87 | NEG | Transaction is terminated | |
| 88 | POS | LPC or PCIe Root Port | RCFG.RPRID = 1 |
| 89 - 8B | NEG | Transaction is terminated | |
| 8C - 8E | POS | LPC or PCIe Root Port | RCFG.RPRID = 1 |

## Table 3-3. Host Fixed I/O Address Space (Sheet 2 of 4)

| Address (hex) | Address Decode | Targets | Register Control |
|---|---|---|---|
| 8F | NEG | Transaction is terminated | |
| 90 | NEG | LPC | |
| 91 | NEG | Transaction is terminated | |
| 92 | NEG | P2SB->ITSS (CPU I/F) | |
| 93 | NEG | Transaction is terminated | |
| 94 - 96 | NEG | LPC | |
| 97 | NEG | Transaction is terminated | |
| 98 | NEG | LPC | |
| 99 - 9B | NEG | Transaction is terminated | |
| 9C - 9E | NEG | LPC | |
| 9F | NEG | Transaction is terminated | |
| A0 - A1 | NEG | 8259 PIC | |
| A4 - A5 | NEG | 8259 PIC | |
| A8 - A9 | NEG | 8259 PIC | |
| AC - AD | NEG | 8259 PIC | |
| B0 - B1 | NEG | 8259 PIC | |
| B2 - B3 | NEG | Power Management - SMI | |
| B4 - B5 | NEG | 8259 PIC | |
| B8 - B9 | NEG | 8259 PIC | |
| BC - BD | NEG | 8259 PIC | |
| C0 - DF | NEG | Transaction is terminated | |
| E0 - FF | NEG | Transaction is terminated | |
| 170 - 177 | NEG | Transaction is terminated | |
| 1F0 - 1F7 | NEG | Transaction is terminated | |
| 200 - 207 | POS | LPC | LPC: IOE.LGE = 1 |
| 208 - 20F | POS | LPC | LPC: IOE.HGE = 1 |
| 220 - 227 | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB |
| 228 - 22F | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB |
| 238 - 23F | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB |
| 278 - 27F | POS | LPC | LPC: IOD.HGE = 1 |

## Table 3-3. Host Fixed I/O Address Space (Sheet 3 of 4)

| Address (hex) | Address Decode | Targets | Register Control |
|---|---|---|---|
| 2E8 - 2EF | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB<br><br>***Note:*** If register is programmed to send to LPC then HSUART will not receive these requests even if programmed to 2E8 |
| | Fabric | HSUART COM4 if programmed by BIOS and locked | |
| 2F8 - 2FF | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB<br><br>***Note:*** If register is programmed to send to LPC then HSUART will not receive these requests even if programmed to 2F8 |
| | Fabric | HSUART COM2 if programmed by BIOS and locked | |
| 338 - 33F | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB |
| 370 - 375 | POS | LPC | LPC:<br>IOD.FDD<br>IOE.FDE |
| 376 | NEG | Transaction is Terminated | |
| 377 | POS | LPC | LPC:<br>IOD.FDD<br>IOE.FDE |
| 378 - 37F | POS | LPC | LPC:<br>IOD.LTP<br>IOE.PPE |
| 3B0 - 3B8 | Fabric | Routed to the PCIe bridge when BCTL.VGAE is set | |
| 3BC - 3BE | POS | LPC | LPC:<br>IOD.LTP<br>IOE.PPE |
| 3C0 - 3DF | Fabric | Routed to the PCIe bridge when BCTL.VGAE is set | |

## Table 3-3. Host Fixed I/O Address Space (Sheet 4 of 4)

| Address (hex) | Address Decode | Targets | Register Control |
|---|---|---|---|
| 3E8 - 3EF | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB<br><br>***Note:*** If register is programmed to send to LPC then HSUART will not receive these requests even if programmed to 2E8 |
| | Fabric | HSUART COM3 if programmed by BIOS and locked | |
| 3F0 - 3F5 | POS | LPC | LPC:<br>IOD.FDD<br>IOE.FDE |
| 3F6 | NEG | Transaction is Terminated | |
| 3F7 - 3F7 | POS | LPC | LPC:<br>IOD.FDD<br>IOE.FDE |
| 3F8 - 3FF | POS | LPC | LPC:<br>IOE.CAE<br>IOD.CA<br>or<br>IOE.CBE<br>IOD.CB<br><br>***Note:*** If register is programmed to send to LPC then HSUART will not receive these requests even if programmed to 2E8 |
| | Fabric | HSUART COM1 if programmed by BIOS and locked | |
| 4D0 - 4D1<br>ITSS<br>(Interrupt) | POS | 8259 PIC | n/a |
| 678 - 67F | POS | LPC | LPC:<br>IOD.LTP<br>IOE.PPE |
| 778 - 77F | POS | LPC | LPC:<br>IOD.LTP<br>IOE.PPE |
| 7BC - 7BE | POS | LPC | LPC:<br>IOD.LTP<br>IOE.PPE |
| CF8<br>*Dword (32-bit) access only* | POS | Access to Configuration Registers | |
| CF9 | POS | Software-Generated Reset | n/a |
| CFC<br>*Dword (32-bit) access only* | POS | Access to Configuration Registers | |

## 3.2.2 Host I/O Space—Relocatable I/O Addresses

### Table 3-4. Host Relocatable I/O Address Space (Sheet 1 of 2)

| Device | Base Address (decimal) | Size | Comment |
|---|---|---|---|
| **PCI Express Root Port Cluster 0** | | | |
| Port 0 (RP0) | B0:D9:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 1 (RP1) | B0:D10:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 2 (RP2) | B0:D11:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 3 (RP3) | B0:D12:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| **PCI Express Root Port Cluster 1** | | | |
| Port 0 (RP4) | B0:D14:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 1 (RP5) | B0:D15:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 2 (RP6) | B0:D16:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port 3 (RP7) | B0:D17:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| **SATA Controller 0** | | | |
| SATA 0 | B0:D19:F0:PCMDBA (BAR0) | 8 Bytes | SATA Controller 0 (AHCI base) |
| | B0:D19:F0:PCTLBA (BAR1) | 4 Bytes | SATA Controller 0 (AHCI base) |
| | B0:D19:F0:SCMDBA (BAR2) | 8 Bytes | SATA Controller 0 (AHCI base) |
| | B0:D19:F0:SCTLBA (BAR3) | 4 Bytes | SATA Controller 0 (AHCI base) |
| | B0:D19:F0:LBAR (BAR4) | 32 Bytes | SATA Controller 0 (AHCI base) |
| **SATA Controller 1** | | | |
| SATA 1 | B0:D20:F0:PCMDBA (BAR0) | 8 Bytes | SATA Controller 1 (AHCI base) |
| | B0:D20:F0:PCTLBA (BAR1) | 4 Bytes | SATA Controller 1 (AHCI base) |
| | B0:D20:F0:SCMDBA (BAR2) | 8 Bytes | SATA Controller 1 (AHCI base) |
| | B0:D20:F0:SCTLBA (BAR3) | 4 Bytes | SATA Controller 1 (AHCI base) |
| | B0:D20:F0:LBAR (BAR4) | 32 Bytes | SATA Controller 1 (AHCI base) |
| **Dedicated PCI Express Root Ports for LAN0 and LAN1** | | | |
| Port for LAN0 | B0:D22:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| Port for LAN1 | B0:D23:F0:IOBASE | Up to IOLIMIT | PCIe Legacy I/O Access |
| **LAN Controllers via Dedicated Root Ports (If in I/O Mode)** | | | |
| LAN 0 | By:D0:Fn:BAR2 | 32 Bytes | Each Primary Function.<br>Bus number assigned during Enumeration |
| LAN 1 | Bz:D0:Fn:BAR2 | 32 Bytes | |

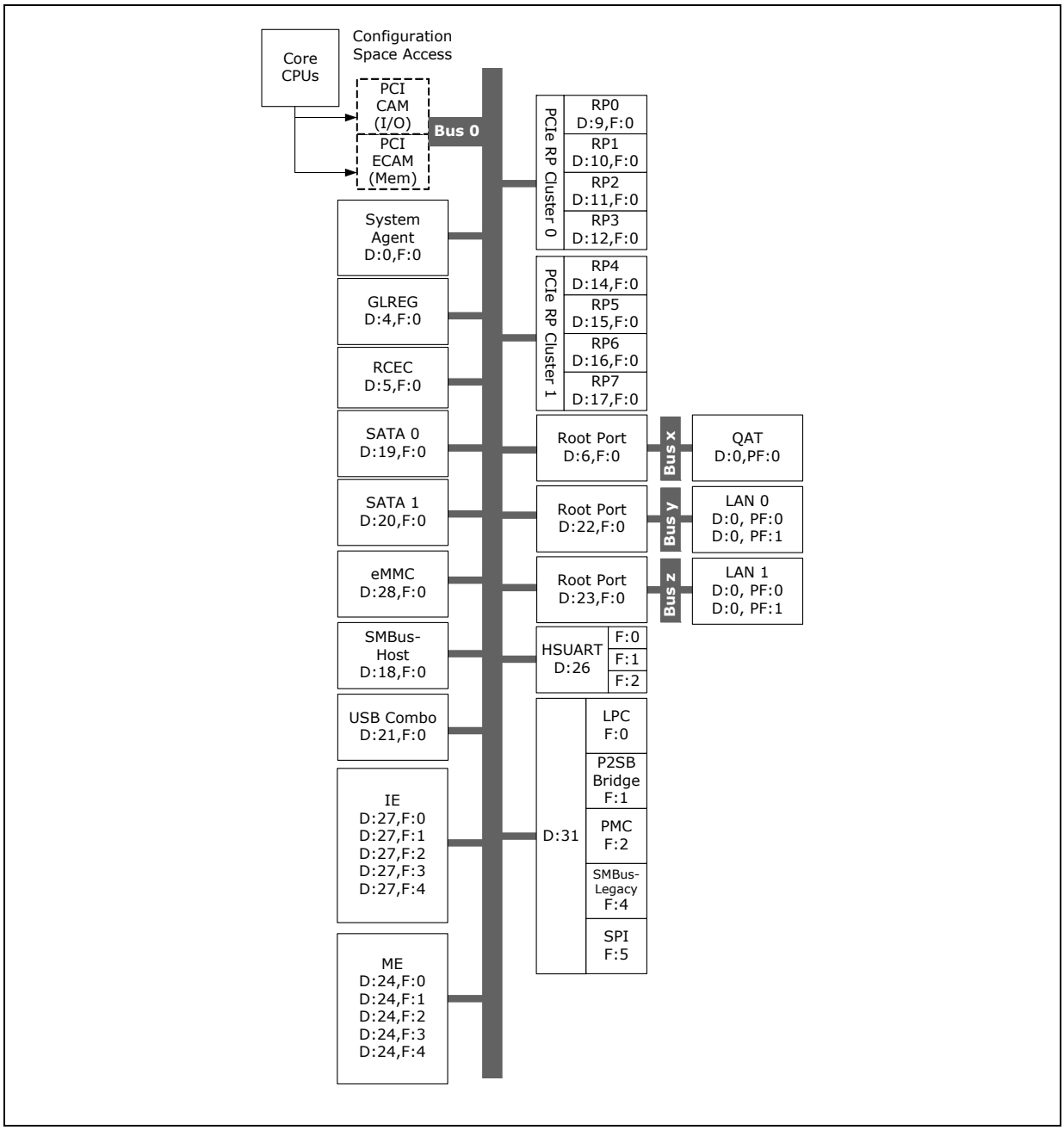
## Table 3-4. Host Relocatable I/O Address Space (Sheet 2 of 2)

| Device | Base Address (decimal) | Size | Comment |
|---|---|---|---|
| **Intel® ME - IDE Redirector (IDER)** | | | |
| Intel® ME - IDE Redirector | B0:D24:F2 IDE_HOST_PCMDIOBAR | 8 Bytes | IDE Primary Command Block |
| | B0:D24:F2 IDE_HOST_PCTLIOBAR | 4 Bytes | IDE Primary Control Block |
| | B0:D24:F2 IDE_HOST_SCMDIOBAR | 8 Bytes | IDE Secondary Command Block (SoC hardware emulates a permanently disabled secondary IDE device). |
| | B0:D24:F2 IDE_HOST_SCTLIOBAR | 4 Bytes | IDE Secondary Control Block (SoC hardware emulates a permanently disabled secondary IDE device). |
| | B0:D24:F2 IDE_HOST_BMIOBAR | 16 Bytes | IDE Bus Master Block |
| **Intel® ME - Keyboard Text (KT)** | | | |
| Intel® ME - KT | B0:D24:F3: KT_HOST_IOBAR | 8 Bytes | Intel® ME - Keyboard Text |
| **High-Speed UART Controllers (In Legacy Mode)** | | | |
| UART 0 | B0:D26:F0:IOBA | 8 Bytes | HSUART |
| UART 1 | B0:D26:F1:IOBA | 8 Bytes | HSUART |
| UART 2 | B0:D26:F2:IOBA | 8 Bytes | HSUART |
| **Innovation Engine (IE) - IDE Redirector** | | | |
| IE - IDE Redirector | B0:D27:F2 IDE_HOST_PCMDIOBAR | 8 Bytes | IDE Primary Command Block |
| | B0:D27:F2 IDE_HOST_PCTLIOBAR | 4 Bytes | IDE Primary Control Block |
| | B0:D27:F2 IDE_HOST_SCMDIOBAR | 8 Bytes | IDE Secondary Command Block (SoC hardware emulates a permanently disabled secondary IDE device). |
| | B0:D27:F2 IDE_HOST_SCTLIOBAR | 4 Bytes | IIDE Secondary Control Block (SoC hardware emulates a permanently disabled secondary IDE device). |
| | B0:D27:F2 IDE_HOST_BMIOBAR | 16 Bytes | IDE Bus Master Block |
| **Innovation Engine (IE) - Keyboard Text (KT)** | | | |
| IE - KT | B0:D27:F3: KT_HOST_IOBAR | 8 Bytes | IE - Keyboard Text |
| **LPC Relocatable Generic I/O Space Regions** | | | |
| LPC | B0:D31:F0:LGIR1 | 256 Bytes | LPC Generic I/O Range #1 |
| | B0:D31:F0:LGIR2 | 256 Bytes | LPC Generic I/O Range #2 |
| | B0:D31:F0:LGIR3 | 256 Bytes | LPC Generic I/O Range #3 |
| | B0:D31:F0:LGIR4 | 256 Bytes | LPC Generic I/O Range #4 |

## 3.3 Host Root Space—PCI Configuration Register Space

Figure 3-1, "Host Root Space - PCI Device Map" shows the PCI map for the Host Root Space. All PCI devices and functions are shown in Table 3-5, "Host PCI Bus, Device, Function Numbers and Device IDs".

**Figure 3-1. Host Root Space - PCI Device Map**



## Table 3-5. Host PCI Bus, Device, Function Numbers and Device IDs (Sheet 1 of 2)

| Bus | Device (dec) | Function | Device ID | Description | Comment |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0x1980 through 0x199F | System Agent | |
| 0 | 4 | 0 | 0x19A1 | GLREG Registers | |
| 0 | 5 | 0 | 0x19A2 | Root Complex Event Collector (RCEC) | |
| 0 | 6 | 0 | 0x19A3 | Dedicated Root Port for Intel® QAT Device | |
| 0 | 9 | 0 | 0x19A4 | PCI Express Cluster 0, Port 0 (RP0) | |
| 0 | 10 | 0 | 0x19A5 | PCI Express Cluster 0, Port 1 (RP1) | |
| 0 | 11 | 0 | 0x19A6 | PCI Express Cluster 0, Port 2 (RP2) | |
| 0 | 12 | 0 | 0x19A7 | PCI Express Cluster 0, Port 3 (RP3) | |
| 0 | 14 | 0 | 0x19A8 | PCI Express Cluster 1, Port 0 (RP4) | |
| 0 | 15 | 0 | 0x19A9 | PCI Express Cluster 1, Port 1 (RP5) | |
| 0 | 16 | 0 | 0x19AA | PCI Express Cluster 1, Port 2 (RP6) | |
| 0 | 17 | 0 | 0x19AB | PCI Express Cluster 1, Port 3 (RP7) | |
| 0 | 18 | 0 | 0x19AC | SMBus Controller - Host | |
| 0 | 19 | 0 | 0x19B0 through 0x19B7, 0x19BE, 0x19BF | SATA Controller 0 | Device IDs 0x19B1 through 0x19B7, 0x19BE, 0x19BF are to be used if the device supports multiple configurations. |
| 0 | 20 | 0 | 0x19C0 through 0x19C7, 0x19CE, 0x19CF | SATA Controller 1 | Device IDs 0x19C1 through 0x19C7, 0x19CE, 0x19CF are to be used if the device supports multiple configurations. |
| 0 | 21 | 0 | 0x19D0 | USB 2.0/3.0 Combo Controller | |
| 0 | 22 | 0 | 0x19D1 | Dedicated Root Port for LAN Controller 0 | |
| 0 | 23 | 0 | 0x19D2 | Dedicated Root Port for LAN Controller 1 | |
| 0 | 24 | 0 | 0x19D3 | Intel® ME - HECI 1 | |
| 0 | 24 | 1 | 0x19D4 | Intel® ME - HECI 2 | |
| 0 | 24 | 3 | 0x19D5 | Intel® ME - Keyboard and Text (KT) Redirection | |
| 0 | 24 | 4 | 0x19D6 | Intel® ME - HECI 3 | |
| 0 | 26 | 0, 1 and 2 | 0x19D8 | High-Speed UART Controllers | |
| 0 | 27 | 0 | 0x19E5 | IE - HECI 1 | |
| 0 | 27 | 1 | 0x19E6 | IE - HECI 2 | |
| 0 | 27 | 3 | 0x19E8 | IE - Keyboard and Text (KT) Redirection | |
| 0 | 27 | 4 | 0x19E9 | IE - HECI 3 | |
| 0 | 28 | 0 | 0x19DB | eMMC | |
| 0 | 31 | 0 | 0x19DC | LPC | |
| 0 | 31 | 1 | 0x19DD | P2SB | |

## Table 3-5. Host PCI Bus, Device, Function Numbers and Device IDs (Sheet 2 of 2)

| Bus | Device (dec) | Function | Device ID | Description | Comment |
|---|---|---|---|---|---|
| 0 | 31 | 2 | 0x19DE | PMC | |
| 0 | 31 | 4 | 0x19DF | SMBus Controller - Legacy | |
| 0 | 31 | 5 | 0x19E0 | SPI Controller | |
| 0 | 31 | 7 | 0x19E1 | Intel® Trace Hub | Details are not provided in this document. |
| x | 0 | PF: 0<br>VF: 0 through 15 | 0x19E2<br>0x19E2 | Intel® QAT - attached to a dedicated Root Port | x = Bus Number assigned during enumeration.<br>PF = Primary Function number<br>VF = Virtual Function numbers |
| y | 0 | PF: 0<br>VF: 0 through 63<br>PF: 1<br>VF: 0 through 63 | *see comment* | LAN Controller 0<br>*attached to a dedicated Root Port* | y, z = Bus Number assigned during enumeration.<br>PF = Primary Function number<br>VF = Virtual Function numbers<br>Device ID:<br>See Table 13-15, "Link Modes" |
| z | 0 | PF: 0<br>VF: 0 through 63<br>PF: 1<br>VF: 0 through 63 | *see comment* | LAN Controller 1<br>*attached to a dedicated Root Port* | |

### 3.3.1 How to Access Registers in PCI Configuration Space

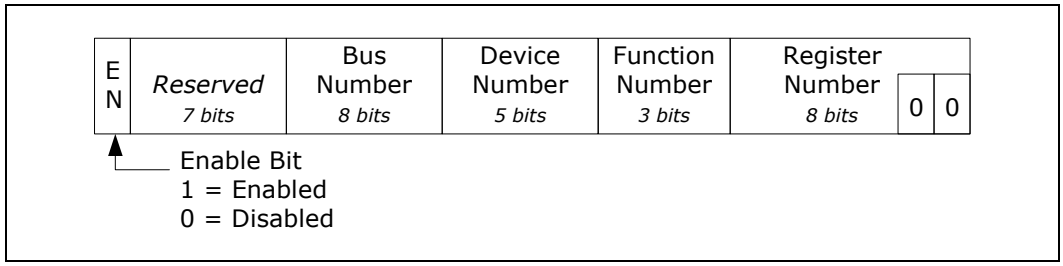
Access to the PCI configuration space registers is performed through one of two Configuration Access Methods (CAMs):

- I/O indexed - PCI compatible Configuration Access Mechanism (CAM)
- Memory mapped - PCI Express Enhanced Configuration Access Mechanism (ECAM)

CAM provides access to the first 256 bytes of the configuration registers for each of the 65,536 possible Functions in a system. It involves performing a DWORD I/O instruction to I/O address CF8h (CONFIG_ADDRESS) followed by another DWORD I/O instruction to I/O address CFCh (CONFIG_DATA). Bit 31 of CF8h must be set to a 1 for a Configuration Request cycle to be generated. CAM is described in the "Software Generation of Configuration Transactions" section of the *PCI Express Base Specification Revision 3.0*.

Figure 3-2, "DWORD Format of CONFIG_ADDRESS at I/O Address CF8h" shows the format of the DWORD for CF8h.

**Figure 3-2. DWORD Format of CONFIG_ADDRESS at I/O Address CF8h**




ECAM was introduced with PCI Express* and extends each function's configuration space to 4,096 bytes. The exact implementation details are product specific, and a general description of the ECAM is provided in the "PCI Express Enhanced Configuration Access" section of *PCI Express Base Specification Revision 3.0*.

The SoC implements ECAM through the Host Root Memory Space. There is a 256-MB relocatable region of memory addresses that provides access to all 4,096 bytes of configuration space for all 65,536 possible functions. This region can be located on a 256-MB memory address boundary within the first 4 GB addresses of Host Memory Space. See Table 3-1, "Host Memory Space Address Map". A 32-bit base memory address is defined by the PCIEXBAR_LO_0_0_0_PCI and PCIEXBAR_HI_0_0_0_PCI registers which are located in Configuration Space at offsets 60h and 64h or Bus 0, Device 0, Function 0.

ECAM utilizes a flat memory-mapped address space to access device configuration registers. This address space is reported by the system firmware to the OS. When using ECAM, in order to maintain compatibility with the PCI configuration addressing mechanisms, system software must access the extended configuration space using 32-bit data operations (32-bit aligned) only. These 32-bit operations include byte enables allowing only appropriate bytes within the DWORD to be accessed. Locked transactions to the PCI Express memory-mapped configuration address space are not supported. All changes made using either the CAM or the ECAM are equivalent.

To access Configuration Space via the ECAM:

1. Use the CAM to enable the ECAM by writing 1 to bit 0 of PCIEXBAR_LO_0_0_0_PCI.
2. Use the CAM to write an appropriate ECAM base address into PCIEXBAR (PCIEXBAR_LO_0_0_0_PCI and PCIEXBAR_HI_0_0_0_PCI).
3. Calculate the memory-mapped address of the Configuration Register to be accessed:
   Memory Address =
   PCIEXBAR +
   Bus Number * 1 MB +
   Device Number * 32 KB +
   Function Number * 4 KB +
   Configuration Register Offset
4. Use a memory write or memory read cycle to the calculated memory-mapped address to write or read the PCI Configuration Register.

If the PCI Bus Number is zero, the SoC generates a Type 0 Configuration Request cycle. If the PCI Bus Number is non-zero, the SoC generates a Type 1 Configuration Request cycle.

## 3.4 Host Root Space - Host Sideband Register Access

### 3.4.1 Using SBREG_BAR Method

The SBREG_BAR access method is safe for SoC product SKUs that have multiple threads. It is the required method for accessing the Host Sideband registers.

The SoC provides access to its Host Sideband registers through a relocatable, 16-MB area of Host Memory Space. The base address is the 64-bit SBREG_BAR (SBREG_BAR, SBREG_BARH) register located in P2SB Bridge's Host Configuration Space B:0, D:31, F:1, offset 10h and 14h. The access mechanism allows direct mapping to 16 bits of byte-addressable register space for each of the integrated Host Sideband registers.

All accesses to the SBREG_BAR region of Host Memory Space are DWord-aligned DWords.

- Writes larger than one DWord are completed normally, but the write data beyond the first DWord is discarded.
- Reads larger than one DWord result in a Unsupported Request (UR) response.
- Byte enables within the DWord are honored normally, enabling accesses of less than one DWord.

### 3.4.2 Accessing System Agent Sideband Registers

BIOS needs to access the B_CR_BMISC_0_0_0_MCHBAR (BMISC) Sideband register of the SoC System Agent in order to set certain access boundaries of Host Memory Space.

To access this and other special System Agent registers, the SoC provides a secure, Memory-Mapped I/O (MMIO) aperture at the 64 bit memory address defined by MCHBAR_LO_0_0_0_PCI and MCHBAR_HI_0_0_0_PCI to expose certain, internal registers of the System Agent. The MCHBAR base address registers are defined in Host Configuration Space at Bus 0, Device 0, Function 0, offset 48h and 4Ch. Not all System Agent Registers are accessible in this aperture.

## 3.5 Key Registers Mentioned in this Chapter

The registers used to configure the system Host Memory-Space map are listed here. This information is provided primarily for BIOS developers.

### 3.5.1 Key Registers in Host Configuration Space

The registers listed in Table 3-6, "Configuration Registers Used to Configure Host Memory Space" are accessed using the CAM method described previously. Once the PCIEXBAR is set using CAM, these registers are also accessible using ECAM.

**Table 3-6. Configuration Registers Used to Configure Host Memory Space (Sheet 1 of 2)**

| PCI Bus, Device, Function (decimal) | Offset (hex) | Register Short Name | Description |
|---|---|---|---|
| **SoC System Agent Registers[1]** | | | |
| B0,D0,F0 | 48 | MCHBAR_LO_0_0_0_PCI | Memory Controller Hub Base Address Register (MCHBAR). Used for accessing certain Host Sideband Registers |
| | 4C | MCHBAR_HI_0_0_0_PCI | |
| | 54 | DEVEN_0_0_0_PCI | Used to enable/hide SoC resources |
| | 60 | PCIEXBAR_LO_0_0_0_PCI | PCI Express Enhanced Configuration Access Mechanism (ECAM) location |
| | 64 | PCIEXBAR_HI_0_0_0_PCI | |
| | A8 | TOUUD_LO_0_0_0_PCI | Top of Upper Usable DRAM TOUUD |
| | AC | TOUUD_HI_0_0_0_PCI | |
| | B0 | BDSM_0_0_0_PCI | Base address of the Data Stolen Memory (DSM) |
| | B4 | BGSM_0_0_0_PCI | Base address of the Graphics Stolen Memory (GSM) |
| | B8 | TSEGMB_0_0_0_PCI | Base address of TSEG DRAM memory |
| | BC | TOLUD_0_0_0_PCI | Top of Lower Usable DRAM (TOLUD) range and start of the Lower MMIO Address range |
| **SPI Controller[2]** | | | |
| B0,D31,F5 | D8 | BIOS_SPI_BDE | BIOS Decode Enable (BDE) register when BIOS is resident on SPI. |
| **LPC Controller[3]** | | | |
| B0,D31,F0 | 84 | LGIR1 | LPC Generic I/O Range #1 |
| | 88 | LGIR2 | LPC Generic I/O Range #2 |
| | 8C | LGIR3 | LPC Generic I/O Range #3 |
| | 90 | LGIR4 | LPC Generic I/O Range #4 |
| | D8 | BDE | BIOS Decode Enable (BDE) register when BIOS is resident on the LPC bus. |

## Table 3-6. Configuration Registers Used to Configure Host Memory Space (Sheet 2 of 2)

| PCI Bus, Device, Function (decimal) | Offset (hex) | Register Short Name | Description |
|---|---|---|---|
| **Primary to Side Band (P2SB) Bridge and Integrated Error Handler (IEH)** | | | |
| B0,D31,F1 | 10 | SBREG_BAR | Lower DWORD of the base address for Host Sideband register access. |
| | 14 | SBREG_BARH | Upper DWORD of the base address for Host Sideband register access. |
| | 54 | RCFG | Routing Configuration |
| | 64 | IOAC | IOxAPIC Configuration |
| **PCI Express Integrated Root Port Cluster 0** | | | |
| B0,D9,F0 | 3E | BCTL | Bridge Control Register - Root Port 0 |
| B0,D10,F0 | 3E | BCTL | Bridge Control Register - Root Port 1 |
| B0,D11,F0 | 3E | BCTL | Bridge Control Register - Root Port 2 |
| B0,D12,F0 | 3E | BCTL | Bridge Control Register - Root Port 3 |
| **PCI Express Integrated Root Port Cluster 1** | | | |
| B0,D14,F0 | 3E | BCTL | Bridge Control Register - Root Port 4 |
| B0,D15,F0 | 3E | BCTL | Bridge Control Register - Root Port 5 |
| B0,D16,F0 | 3E | BCTL | Bridge Control Register - Root Port 6 |
| B0,D17,F0 | 3E | BCTL | Bridge Control Register - Root Port 7 |

1. The System Agent automatically shadows these accesses to certain other SoC registers. These registers must not be accessed by BIOS via sideband.
2. The sampled Boot BIOS Strap (BBS) hard strap determines whether the BIOS Decode Enable is taken from the SPI Controller function B0,D31,F5 or from the LPC function B0,D31,F0.
3. The sampled LPC Decode Select hard strap determines whether the B0,D31,F0 BIOS Decode Enable belongs to the LPC Controller.

## 3.5.2 Key Registers in Host Memory Space

### Table 3-7. Relocatable Host Memory-Mapped Aperture to Access Certain System Agent Registers

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Description |
|---|---|---|---|
| MCHBAR_LO_0_0_0_PCI, MCHBAR_HI_0_0_0_PCI | 6800 | B_CR_BMISC_0_0_0_MCHBAR | Miscellaneous Configuration Register. BIOS must use MCHBAR to program BMISC. |
| | 6C80 | B_CR_DEFVTDBAR_0_0_0_MCHBAR | Default VTd Base Address |

## 3.5.3 Key Registers in Host I/O Space

### Table 3-8. Fixed-Address Host I/O-Mapped Registers to Access Configuration Space

| I/O Address (hexadecimal) | Register Short Name | Description |
|---|---|---|
| CF8 | CONFIG_ADDRESS | DWORD containing the Configuration Register address. See Figure 3-2, "DWORD Format of CONFIG_ADDRESS at I/O Address CF8h". |
| CFC | CONFIG_DATA | DWORD containing the Configuration Register data. |

§ §

# 4 Strapping and Configuration

This chapter describes how to configure the SoC through the use of hard and soft straps and provides descriptions on what the different strapping options do. This chapter will cover two kinds of strapping configurations: hard straps or pin straps and soft straps. This chapter will also briefly address in-field programmable fuses pertaining to the Innovation Engine (IE) for more information please refer to Chapter 37, "Innovation Engine."

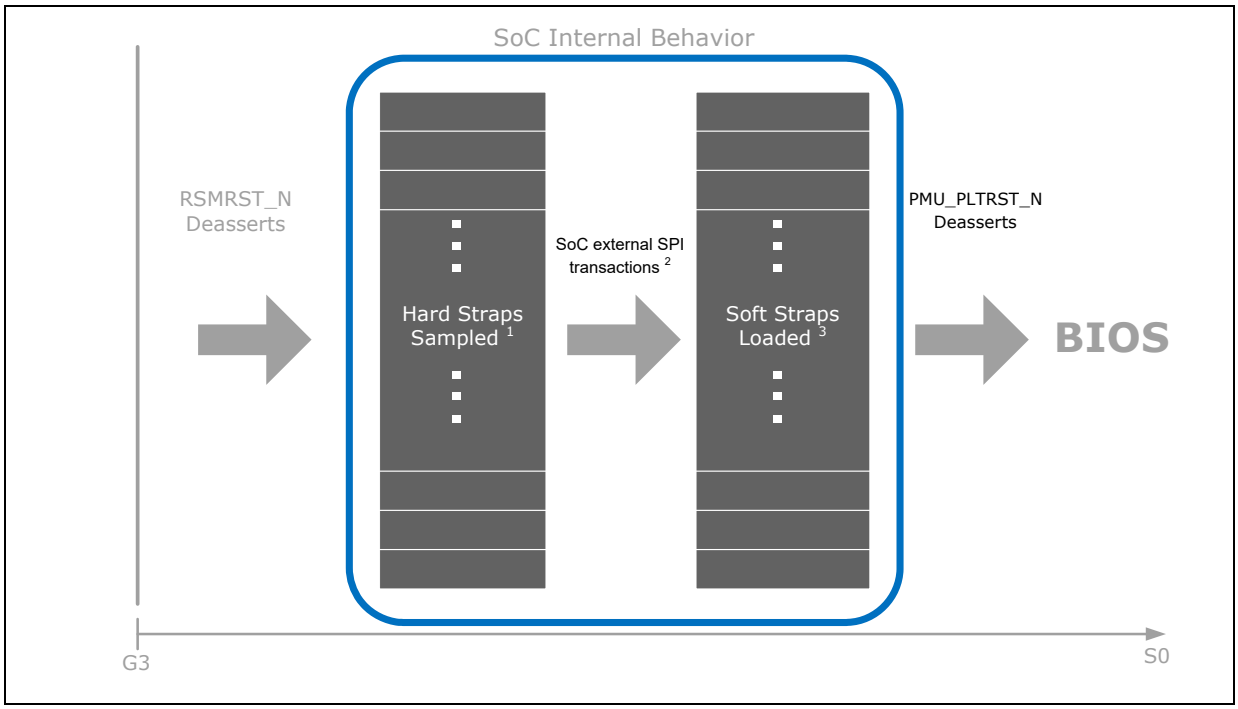
Strapping is needed to enable, disable, or alter default configurations in the SoC. This chapter will provide the descriptions of these strapping signals, their functionality and the configurations they support. For Soft Straps, please refer to Section 4.2, "SoC Soft Straps" in this chapter.

For this SoC, hard straps are the earliest form of user-defined configuration available in the SoC boot sequence. Hard straps are connected to a physical ball or pin depending on the SoC's package and thus are typically very few in number. Some hard straps, will drive a separate function such as a GPIO from the same SoC ball or pin after the hard strap sampling period, which ends 400 ns after the deassertion of the signal RSMRST_N. If hard straps are not set properly, a user is likely to see unexpected complications while trying to boot the SoC. This includes the possibility of no code being fetched from the boot flash memory device since hard straps are responsible for configuring which boot interface will be used by the SoC. It is recommended that customers take precautions and allow themselves the flexibility to easily configure hard straps in their designs through the use of jumpers or other means when first starting a design and are still performing their design validation with the SoC. This allows customers to easily reconfigure hard straps and avoid potentially strenuous reworks to their design. A customer can choose to remove these jumpers or other means for the production version of their design to save on BOM cost. Please refer to Section 4.1, "SoC Hard (Pin) Straps" of this chapter for more information on configuring hard straps.

Soft straps are the second earliest form of user-defined configuration available in the SoC boot sequence and unlike hard straps they are numerous. This SoC has over 2000 soft strap bits which live in the descriptor region of the boot flash memory device. Some soft straps rely on one soft strap bit to configure one setting in the SoC, other soft straps require four or more soft strap bits to configure one setting in the SoC. This chapter will refer to a soft strap as a single or multiple soft strap bits that are required to configure one SoC setting and are listed in Table 4-3, "Flash Descriptor Soft Straps." To be able to load the soft straps into the SoC, it is necessary to read them from the boot flash memory device connected to the SoC which will trigger transactions between the SoC and the device where the soft straps are stored at. i.e. if using a SPI flash memory device to store the soft straps, the user should be able to see and monitor the SPI transactions happening between the SoC and the SPI flash memory device when the SoC is reading the soft straps from the SPI device. Soft straps are required for many of the SoC controllers to come out of reset and as such it is necessary that soft straps are fetched by the SoC in order to continue with the SoC boot sequence. If no soft straps are found by the SoC, the SoC will stop its power on sequence before the signal PMU_SLP_S45_N deasserts. Soft Straps are fetched before the reset vector fetches the first BIOS instruction.

Figure 4-1, "Strap Loading Flow" shows that the strap loading flow starts with the deassertion of the reset signal RSMRST_N, after which the hard straps are sampled followed by the soft straps. The SoC requires that the platform SUSCLK is stable before soft straps are fetched from the boot flash memory device attached to the SoC through the SPI interface. Additionally, the SoC needs to finish reading soft straps before PMU_SLP_S45_N deasserts or the SoC will not progress further in its power on sequence. After PMU_PLTRST_N deasserts, the board follows the boot sequencing flow to the BIOS.

**Figure 4-1. Strap Loading Flow**




***Notes:***

1. Hard straps are responsible for configuring which boot interface will be used by the SoC and hence where the SoC will try to read the soft straps from. Please refer to table Table 4-2, "Boot interface Hard Strap/ Pin Strap Configurations" for more information.
2. Platform SUSCLK must be stable before SPI transactions are initiated.
3. Soft Straps live in the descriptor region of the boot memory flash device that can reside over the SPI interface. The SoC needs to finish reading soft straps before the signal PMU_SLP_S45_N deasserts or the SoC will not progress further in its power on sequence.

## 4.1 SoC Hard (Pin) Straps

Hard straps are configuration signals connected to a physical ball or pin in the SoC depending on the SoC packaging, hence sometimes hard straps are also called pin straps. Since hard straps are connected to a single ball or pin, each hard strap configures a different SoC setting and can only have one of two values: 0 (low) or 1 (high). In order to configure the SoC hard strap to a 1 (high), a pull up is required. In order to configure the SoC hard strap to a 0 (low), a pull down is required. To configure a hard strap pull up (PU) or pull down (PD), a customer can choose to use the SoC internal hard strap default PU/PD or they can force the hard strap with an external PU/PD implemented on the platform.

When configuring hard straps with the SoC internal PU/PD, there is no need to connect the hard strap ball or pin to any termination on the board, it can be left floating. However, there is no way to configure the SoC internal hard strap from a PU to a PD or vice-versa, so the hard strap default will always be either a PU or a PD. See Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions" to check the internal SoC hard strap default PU/PD. When a customer does not require the hard strap setting associated with the default internal SoC PU/PD, an external PU resistor tied to a voltage rail or a PD resistor tied to ground is required to be used on the platform.

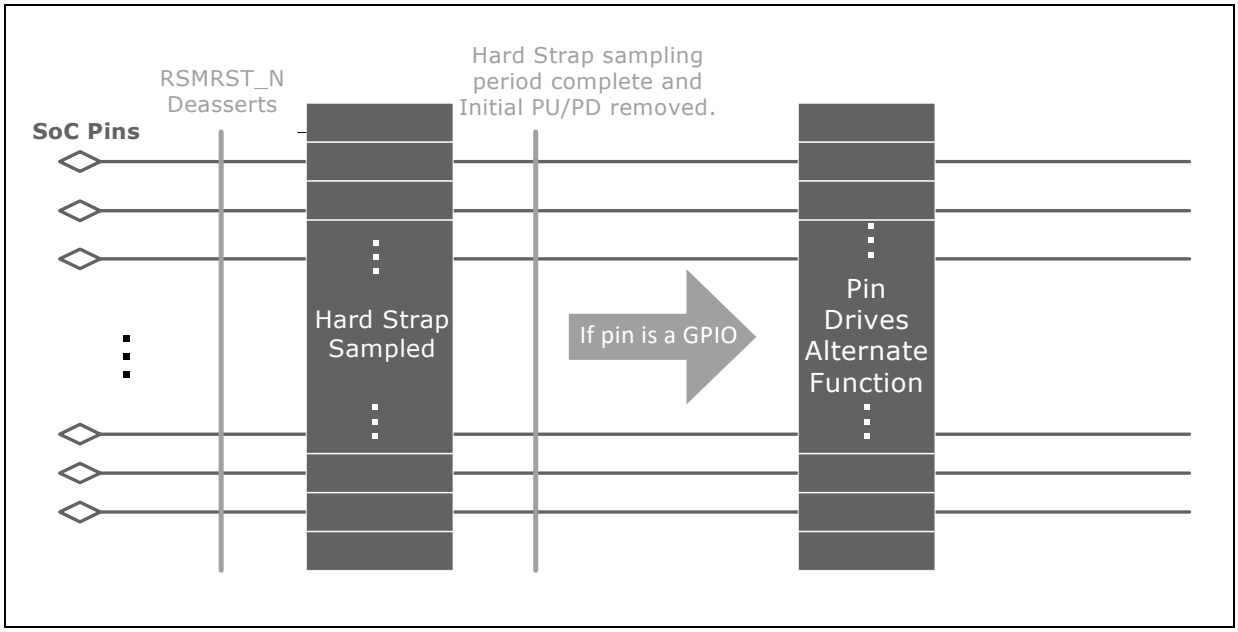
When configuring hard straps with an external PU/PD implemented on the board, the internal SoC PU/PD has to be forced to a new state by the external PU/PD. It is recommended that customers take precautions and allow themselves the flexibility to easily configure hard straps in their designs though the use of jumpers or other means when first starting a design and are still performing their design validation with the SoC. This allows to easily reconfigure hard straps and avoid potentially strenuous reworks to the design. These jumpers or other means can be removed for the production version of the design to save on BOM cost.

Each of the hard strap pins is briefly configured as an SoC input during system power on and its state is sampled during the hard strap sampling period which ends 400 ns after the deassertion of the resume reset signal RSMRST_N; the state of the SoC pin or ball is subsequently latched by the SoC hardware. The platform SUSCLK does not need to be stable for hard straps to be sampled.

After the hard strap sampling period, the internal SoC PU/PD associated with the hard straps are removed and if the SoC pin or ball is also a GPIO, then the SoC will drive the GPIO as the alternate function. The GPIO pin should not be driven by an external device during the hard strap sampling period. Customer should take precautions to ensure this does not happen during the SoC power on sequence. If an external PU/PD was used on the hard strap and the pin is used as a GPIO after the hard strap sampling period, then the external PU/PD on the platform will also affect the GPIO signal. For more information on the hard strap pins refer to Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions" for alternate functions refer to Section 39.1, "Directory of Signal Names and Pin Names". For an illustration see Figure 4-2, "Hard Strap Functionality".

Hard straps are responsible for configuring which boot interface will be used by the SoC to read soft straps and fetch BIOS code from. If hard straps are set incorrectly, soft straps will not be fetched and the SoC will not boot. Please refer to Table 4-2, "Boot interface Hard Strap/ Pin Strap Configurations" for more information.

**Figure 4-2. Hard Strap Functionality**




*Note:* It is important to ensure that pin straps are always connected to the correct voltage rail when pulled up. Most hard straps are tied to the V3P30 rail Voltage Group, with the exception of "Top Swap Override" Hard Strap which is in GPIO[3] Pin and should be tied to V1P8 Voltage Rail. Also the if the GPIO pin is connected to an external device, that external device should not drive the hard strap for at least the duration of the hard strap sampling period.

In Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions", hard strap pins default to inputs on power up. Their hard strap value is sampled within 400 ns of the rising edge (0 -> 1) of the active-low resume reset signal RSMRST_N. Most hard straps are tied to the V3P30 rail Voltage Group, with the exception of "Top Swap Override" Hard Strap which is tied to pin GPIO[3] and should be tied to V1P8 Voltage Rail.

## Table 4-1. SoC Hard Straps/Pin Straps and Descriptions (Sheet 1 of 3)

| Hard Strap | Pin Name | SoC Ball Location | Internal PU/PD[1] | Description |
|---|---|---|---|---|
| Boot BIOS Strap (BBS) | ERROR_N[1] | BR58 | 20K PD | This Hard Strap tells the SoC from which interface the BIOS code will be fetched<br>• 0 = SPI<br>• 1 = LPC<br><br>***Note:*** Please see table Table 4-2, "Boot interface Hard Strap/ Pin Strap Configurations" for information on how to set the Boot BIOS Strap (BBS), LPC Decode Select hard straps properly. |
| LPC Decode Select | ERROR_N[0] | BL52 | 20K PD | This Hard Strap tells the SoC to use the LPC interface.<br>• 0 = Low Pin Count (LPC)<br>• 1 = Reserved<br><br>***Note:*** The system board must supply a 3.3V voltage to the VCC_LPC_ESPI_3P3_1P8 rail in order for the interface to work. Connecting to the 1.8V rail is not supported.<br>***Note:*** Please see table Table 4-2, "Boot interface Hard Strap/ Pin Strap Configurations" for information on how to set the Boot BIOS Strap (BBS), LPC Decode Select hard straps properly. |
| Reserved for Intel | MCERR_N | BN52 | 20K PD | This Hard Strap must always be set to 0. Either through internal PD or an external PD. |
| Flash Security Override | IERR_N | BW50 | 20K PD | • 0 = Flash descriptor security locked (default). The security measures defined in the Flash Descriptor will be in effect.<br>• 1 = Flash descriptor security unlocked<br><br>***Notes:***<br>1. The internal pull-down is disabled after PLTRST_N deasserts.<br>2. Asserting Flash Security Override Hard Strap high on the rising edge of PWROK will also halt Intel ME after SoC bring up and disable runtime Intel ME features. This is a debug mode and must not be asserted after manufacturing/debug.<br>3. This signal is in the Suspend well (V3P30).<br>4. This hard strap might be named MFG_MODE_STRAP on other IA products. |
| No TCO Reboot Enable | GPIO[12] | BN56 | 20K PD | This Hard Strap tells the SoC to enable/disable the Total Cost of Ownership (TCO) watch dog timer (WDT) system reboot feature after the second WDT timeout.<br><br>• 0 = Default State (the system will reboot after second WDT timeout)<br>• 1 = Stop the TCO watchdog timer from rebooting the system after the second timeout.<br><br>***Note:*** More information on TCO see Section 16.6, "Total Cost of Ownership (TCO)". |

## Table 4-1. SoC Hard Straps/Pin Straps and Descriptions (Sheet 2 of 3)

| Hard Strap | Pin Name | SoC Ball Location | Internal PU/PD[1] | Description |
|---|---|---|---|---|
| Top Swap Override | GPIO[3] | BN44 | 20K PD | • 0 = Top-swap feature is disabled.<br>• 1 = Top-swap feature is enabled.<br><br>***Note:*** For more information on Top-Swap override see Section 56.2.12, "BIOS Control (BC)—Offset DCh" on page 2657. |
| Forces ME into Recovery Mode | ME_RCVR_N | BE44 | 20K PU | This signal can be configured via the FITc tool to be on 147 GPIO pins or not available. The default is pin GPIO148.<br>This signal has an internal 20K pull-up to voltage 1.05V with a jumper option to a 1K pull-down resistor on platform.<br>• 0 = Recovery mode<br>• 1 = Normal operating mode for Intel ME.<br>The default for production systems must be normal operating mode.<br><br>***Note:*** This signal will immediately put the ME in recovery mode when driven low. By default, driving the signal high will put the ME back in operational mode. A FITc configuration setting can be used to configure the system for legacy mode where operational mode will not be entered until the next reset. |
| Reserved for Intel | RSVD_BY58 | BY58 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | ERROR_N[2] | CA50 | 20K PD | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | LAN0_PORT0_SDP[3] | BA69 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | LAN0_PORT1_SDP[3] | W58 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | GPIO[0] | BF70 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | GPIO[1] | BK69 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | GPIO[2] | BK71 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |

## Table 4-1. SoC Hard Straps/Pin Straps and Descriptions (Sheet 3 of 3)

| Hard Strap | Pin Name | SoC Ball Location | Internal PU/PD[1] | Description |
|---|---|---|---|---|
| Reserved for Intel | SATA0_SDOUT | BU57 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | SMB_LAN_ALRT_N | BN66 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | SUS_STAT_N | BV58 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | FLEX_CLK_SE[0] | BG48 | 20K PD | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | FLEX_CLK_SE[1] | CA48 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | UART0_TXD | CA52 | 20K PD | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |
| Reserved for Intel | UART1_TXD | BN59 | 20K PU | This signal has to be sampled in its default "Internal PU/PD" state.<br><br>Platform cannot drive this signal until > 400ns after RSMRST_N de-asserts. |

***Notes:***
1. The internal PU/PD is only applied during the hard strap sampling period.
2. Strap sampling period is 400 ns after the deassertion of the RSMRST_N signal.

In order for the SoC to properly fetch BIOS code from the flash storage device over the SPI interface the hard strap Reserved for Intel (MCERR_N) has to be tied low (0) even if the LPC Decode Select (ERROR_N[0]) hard strap is tied low (0) which selects the LPC interface.

Please refer to Table 4-2, "Boot interface Hard Strap/ Pin Strap Configurations" for guidance on configuring the SoC boot interface and where the SoC will fetch BIOS code from. Failure to follow the hard strap configurations in the table below may cause the SoC to not come out of reset.

## Table 4-2. Boot interface Hard Strap/ Pin Strap Configurations

| | | Hard Straps | |
|---|---|---|---|
| **Boot interface** | **LPC Enabled** | **Boot BIOS Strap (ERROR_N[1]) Internal 20K PD** | **LPC Decode Select (ERROR_N[0]) Internal 20K PD** |
| **SPI** | **LPC** | 0 | 0 |
| **LPC** | **LPC** | 1 | 0 |

## 4.2 SoC Soft Straps

Soft straps are used to set and configure various features and configurations. Soft strap information is stored within the flash descriptor (region 0) of the Serial Peripheral Interface (SPI) firmware image (which resides on the SPI interface) and removes the need to use pull-up/pull-down resistors to hard strap most SoC features. At boot, the SPI controller reads the soft strap content from the SPI Flash and then distributes it to the various SoC controllers.

Soft Straps are configured utilizing the Intel FITc tool that is provided in the latest Intel® Server Platform Services Firmware Release for Harrisonville Microserver Platforms kit. It is recommended to use the latest version of the tool since that every time the kit is updated, so is the FITc tool.

The SoC provides the ability to use any of the 20 high speed I/O (HSIO) lanes as long as the number of HSIO lanes enabled is lesser than or equal to the number of HSIO lanes allowed on the SKU via fuses. For more information on how this flow works, see Figure 10-3, "Intel® ME-BIOS/UEFI Interactions in a System with Intel® SPS Firmware". HSIO lanes are configured via soft straps and allow the user to select the SoC controller that the HSIO lane will be connected to. If the HSIO lane is configured to either PCIe Gen 3/ SATA Gen 3/ USB3.0 then the HSIO lane will count as being used. However, if it is desired to disable a particular HSIO lane so that the HSIO lane will not count toward the maximum HSIO lane limit imposed by the SKU, then the soft strap configuration setting of "0x2; HSIO Lane not used" must be programed into the HSIO Lane Assignment soft straps. In addition to configuring the SoC FIA via the HSIO Lane Assignment soft straps to select the number of HSIO lanes used and the SoC controllers they connect to, it is also necessary to configure the individual SoC controllers via soft straps to ensure proper SoC behavior. Please follow the following guidance:

- For each HSIO lane assigned to a SATA controller in the HSIO Lane Assignment soft strap section, configure the respective SATA controller to receive the HSIO lane via the HSIO Lane Configuration from SATA Controller section.
- For each HSIO lane assigned to a PCIe root port cluster in the HSIO Lane Assignment soft strap section, configure the respective PCIe root port cluster to support the desired PCIe configuration by configuring settings such as enabling PCIe root ports in the PCIe Bifurcation and Settings section. Once a PCIe root port is disabled via soft strap, it cannot be enabled via BIOS or registers, soft straps take priority.
- For each SoC HSIO controller being used, make sure that it is enabled via the soft straps found in the SoC Feature Disable. Once an SoC controller is disabled via soft strap, it cannot be turned back on unless the soft strap region is updated in the descriptor region of the boot memory flash device and an A/C power cycle is performed on the system.

Some of the SoC soft straps can also be configured without the need to flash the boot flash memory device with a new soft strap configuration in the descriptor region. Instead BIOS is able to request the ME over a HECI message that certain soft straps be reconfigured after a global reset when booting the SoC. For information on how this flow works, please see Figure 10-3, "Intel® ME-BIOS/UEFI Interactions in a System with Intel® SPS Firmware." The soft straps that are able to be dynamically updated by this feature are limited to the following:

- FIA Mux configuration - HSIO Lane Assignment soft straps
- HSIO lane configuration for SATA controllers - HSIO Lane Configuration from SATA Controller soft straps
- PCIe root port cluster configuration (PCIe Bifurcation and Settings) - PCIe* Root Port Cluster [0-1] - "Root Port [0-7] Enable" soft straps
- PCIe root port cluster configuration (PCIe Bifurcation and Settings) - PCIe* Root Port Cluster [0-1] - "Root Port [0-7] x1 Slot Width" soft straps

*Note:* If no soft straps are found by the SoC, the SoC will stop its power on sequence before PMU_SLP_S45_N deasserts. Soft Straps are fetched before the reset vector fetches the first BIOS instruction.

The Flash Descriptor Soft Straps Table 4-3, "Flash Descriptor Soft Straps" are grouped by SoC features rather than the numerical order of soft straps in the descriptor region of the SPI Flash image.

### Table 4-3. Flash Descriptor Soft Straps (Sheet 1 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Power Delivery** | | | | | | |
| 1 | + 4h | 15 | DRAM VDDQ Voltage Rail - SVID Support | 0x0; SVID for VDDQ Disabled<br>0x1; SVID for VDDQ Enabled | 0x1 | Tells the SoC if the VDDQ voltage rail supports Serial Voltage Identification (SVID) via SVID_CLK, SVID_DATA and SVID_ALERT pins. If SVID is disabled then the VDDQ rail is static. |
| 2 | + 8h | 0 | LAN Controller 0 - Powered in S5 Enable | 0x0; Not powered in S5 WoL not supported<br>0x1; Powered in S5 WoL possible | 0x1 | Tells the SoC that LAN Controller 0 is powered in S5 power state which allows Wake on LAN (WoL) support.<br>**Note:** Setting this strap to "0x0; Not powered in S5; WoL not supported" will cause LAN Controller 0 and 1 to not support WoL regardless of the "LAN Controller 1 Powered in S5 Enable" soft strap setting. |
| 3 | + Ch | 0 | LAN Controller 1 - Powered in S5 Enable | 0x0; Not powered in S5 WoL not supported<br>0x1; Powered in S5 WoL possible | 0x1 | Tells the SoC that LAN Controller 1 is powered in S5 power state which allows Wake on LAN support<br>**Note:** LAN Controller 0 Powered in S5 must be set to "Powered in S5; WoL possible" for LAN Controller 1 to function in S5 |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 2 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Power Sequence and Reset - PMC Knobs and Settings** | | | | | | |
| 4 | + 10h | 9:8 | t573 Timing: COREPWROK to PMU_PLTRST_N Delay | 0x0; 100 ms (default)<br>0x1; 50 ms<br>0x2; 5 ms<br>0x3; 1 ms | 0x0 | Tell the SoC to control timing t573 which is the minimum timing from the assertion of COREPWROK and the de-assertion of the PMU_PLTRST_N signal<br><br>**Note:** The default setting of 100ms is chosen to be conservative, but the typical preferred setting is expected to be 1ms. |
| 6 | + 18h | 11:8 | IE FW Launch Time - Out Multiplier | 0x0; 8 sec<br>0x1; 16 sec<br>0x2; 24 sec<br>0x3; 32 sec<br>0x4; 40 sec<br>0x5; 48 sec<br>0x6; 56 sec<br>0x7; 64 sec<br>0x8; 72 sec<br>0x9; 80 sec<br>0xA; 88 sec<br>0xB; 96 sec<br>0xC; 104 sec<br>0xD; 112 sec<br>0xE; 120 sec<br>0xF; 128 sec | 0x0 | Tells the SoC the watch dog timer time before assuming the IE FW is hung on boot and the system needs to be reset.<br>**Note:** Soft Strap "SS6 bit [17] - IE FW Host Boot Preparation Enable" has to be enabled for this strap to have any effect. |
| 6 | + 18h | 16 | AG3E Boot Select | 0x0; System will boot to S0.<br>0x1; System wait for wake event (S4/S5). | 0x1 | Tells the SoC to return the system to S0 or S4/S5 power states after power is applied for the first time. Refer to Section 16.10.2, "Initiating State Changes when in the G3 Mechanical-Off State" for additional information about G3 transitions to the S0 and S5 states.<br>**Note:** If the system lost power in S4, then it will return to an S4-like state.<br>In the S5 state, power button or any enabled wake events that were preserved through the power failure are valid wake events and will return the system to S0. See Chapter 16, "Initiating State Changes when in the G3 Mechanical-Off State". |
| 6 | + 18h | 17 | IE FW Host Boot Preparation Enable | 0x0; Do not wait for IE FW<br>0x1; Wait for IE FW | 0x0 | Tells the SoC to wait / do not wait for positive indication from IE FW before proceeding with Host boot. |
| 6 | + 18h | 29 | PMU_RESETBUTTON_N De-bounce | 0x0; 16-ms de-bounce<br>0x1; No de-bounce | 0x0 | Tells the SoC to de-bounce the PMU_RESETBUTTON_N pin for 16-ms. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 3 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **SoC Feature Disable** | | | | | | |
| 5 | + 14h | 0 | LAN Controller 0 Disable | 0x0; LAN Ctrl 0 Enabled<br>0x1; LAN Ctrl 0 Disabled | 0x0 | Tells the SoC to disable LAN Controller 0.<br>**Note:** Setting this strap to "LAN Controller 0 Disabled" will cause LAN Controller 1 not to function. |
| 5 | + 14h | 1 | LAN Controller 1 Disable | 0x0; LAN Ctrl 1 Enabled<br>0x1; LAN Ctrl 1 Disabled | 0x0 | Tells the SoC to disable LAN Controller 1.<br>**Note**: LAN Controller 0 Disable strap must be set to "LAN Ctrl 0 Enabled" for LAN Controller 1 to function. |
| 5 | + 14h | 2 | SATA Controller 0 Disable | 0x0; SATA Ctrl 0 Enabled<br>0x1; SATA Ctrl 0 Disabled | 0x0 | Tells the SoC to disable SATA Controller 0.<br>**Note:** Disabling SATA Controller 0 has no effect on SATA Controller 1 functionality.<br>If set to "SATA Ctrl 0 Disabled" set the soft straps "HSIO Lane 4-11 Select" to a setting that is NOT the "SATA" setting.<br>Also set soft straps "SATA Controller 0 - Assign HSIO Lane 4-11 to Port 0-7" to setting "0x1 - HSIO Lane Not assigned to SATA Controller 0"<br>If the SoC SATA is used then one of the following soft straps must also be set to enabled:<br>SS 5 bit [4] PCIe Root Port Cluster 0 Disable OR SS 5 bit [5] PCIe Root Port Cluster 1 Disable OR SS 5 bit [7] USB Combo Controller Disable AND the requirements in this SATA NOTE must be met. |
| 5 | + 14h | 3 | SATA Controller 1 Disable | 0x0; SATA Ctrl 1 Enabled<br>0x1; SATA Ctrl 1 Disabled | 0x0 | Tells the SoC to disable SATA Controller 1.<br>**Note:** Disabling SATA Controller 1 has no effect on SATA Controller 0 functionality.<br>If set to "SATA Ctrl 0 Disabled" set the soft straps "HSIO Lane 12-19 Select" to a setting that is NOT the "SATA" setting.<br>Also set soft straps "SATA Controller 1 - Assign HSIO Lane 12-19 to Port 0-7" to setting "0x1 - HSIO Lane Not assigned to SATA Controller 1"<br>If the SoC SATA is used then one of the following soft straps must also be set to enabled:<br>SS 5 bit [4] PCIe Root Port Cluster 0 Disable OR SS 5 bit [5] PCIe Root Port Cluster 1 Disable OR SS 5 bit [7] USB Combo Controller Disable AND the requirements in this SATA NOTE must be met. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 4 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 5 | + 14h | 4 | PCIe Root Port Cluster 0 Disable | 0x0; PCIe Root Port Cluster 0 Enabled<br>0x1; PCIe Root Port Cluster 0 Disabled | 0x0 | Tells the SoC to disable PCIe Root Port Cluster 0.<br>**Note:** Disabling PCIe Root Port Cluster 0 has no effect on PCIe Root Port Cluster 1 functionality.<br>If set to "PCIe Root Port Cluster 0 Disabled" set the soft straps "HSIO Lane 4-7 Select" to a setting that is NOT the "PCIe" setting. HSIO Lanes 0-3 will be unusable and should be set to setting "0x2 - HSIO Lane not used" in the soft straps "HSIO Lane 0-3 Select". |
| 5 | + 14h | 5 | PCIe Root Port Cluster 1 Disable | 0x0; PCIe Root Port Cluster 1 Enabled<br>0x1; PCIe Root Port Cluster 1 Disabled | 0x0 | Tells the SoC to disable PCIe Root Port Cluster 1.<br>**Note:** Disabling PCIe Root Port Cluster 1 has no effect on PCIe Root Port Cluster 0 functionality.<br>If set to "PCIe Root Port Cluster 1 Disabled" set the soft straps "HSIO Lane 8-15 Select" to a setting that is NOT the "PCIe" setting. |
| 5 | + 14h | 6 | Intel® QuickAssist Technology Disable | 0x0; Intel® QAT Enabled<br>0x1; Intel® QAT Disabled | 0x0 | Tells the SoC to disable the Intel® QuickAssist Technology.<br>**Note:** This strap has no effect if your SKU does not support Intel® QuickAssist Technology. |
| 5 | + 14h | 7 | USB Combo Controller Disable | 0x0; USB Combo Controller Enabled<br>0x1; USB Combo Controller Disabled | 0x0 | Tells the SoC to disable the USB Combo Controller (USB 2.0 + USB 3.0).<br>**Note:** If set to "USB Combo Controller Disabled" set the soft straps "HSIO Lane 16-19 Select" to a setting that is NOT the "USB 3.0" setting. |
| 5 | + 14h | 8 | eMMC Controller Disable | 0x0; eMMC Enabled<br>0x1; eMMC Disabled | 0x0 | Tells the SoC to disable the eMMC (Embedded Multi-Media Card) Controller. |
| 5 | + 14h | 17 | LAN Controller 0 - Port 1 Disable | 0x0; LAN Controller 0 Port 1 Enabled<br>0x1; LAN Controller 0 Port 1 Disabled | 0x0 | Tells the SoC to disable Port 1 of LAN Controller 0.<br>**Note**: This assumes that the "LAN Controller 0 Disable" soft strap is set to "LAN Ctrl 0 Enabled". |
| 5 | + 14h | 19 | LAN Controller 1 - Port 1 Disable | 0x0; LAN Controller 1 Port 1 Enabled<br>0x1; LAN Controller 1 Port 1 Disabled | 0x0 | Tells the SoC to disable Port 1 of LAN Controller 1.<br>**Note:** Enabling LAN Controller 1 assumes LAN controller 1 is enabled in your SKU and the "LAN Controller 0/1 Disable" soft straps is set to "LAN Ctrl 0/1 Enabled". |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 5 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **SoC Boot DDR Frequency** | | | | | | |
| 7 | + 1Ch | 7:4 | SoC Boot DDR Frequency | 0x0; Use SoC Fuse Value<br>0x9; 1866<br>0xA; 2133<br>0xB; 2400<br>**Note:** All other combinations are Reserved | 0x0 | Limits the speed of the SoC memory controlled during boot. This can be used to avoid SPD reset during MRC training. |
| **BMC and BMC-less Mode Error logging** | | | | | | |
| 5 | + 14h | 20 | SMB TCO Watch Dog Timer Disable | 0x0 -BMC-less Mode: TCO WDT is enabled and any WDT timeout would be escalated to a global reset. If SS7 bit 1 is set to enable, then the error causing the WDT timeout would be logged in SUS_SCRATCH_1 register bit 0.<br>0x1 -BMC Mode: Prevents TCO WDT from escalating to a global reset due to errors in internal SoC IPs. This allows the BMC to be able to collect error data before global reset. Errors can be fetched from the integrated error handler (IEH). | 0x0 | Tells the SoC to disable the SMB TCO Watch Dog Timer.<br><br>**Note:** This SMB TCO WDT is internal to PMC and is intended to be used in conjunction with SS7 bit [1] - "BMC and BMC-less Mode SUS_SCRATCH_1 bit[0] Error logging Enable" soft strap to help capture data should a crash happen under system configurations with or without BMC. |
| 7 | + 1Ch | 1 | BMC and BMC-less Mode SUS_SCRATCH_1 bit[0] Error logging Enable | 0x0 - SUS_SCRATCH_1 bit[0] Error logging Disabled.<br>0x1 -SUS_SCRATCH_1 bit[0] Error logging Enable. | 0x0 | When this bit is set, periodically check the status of the IEH and the RCC to check for errors that are impeding a warm reset. If errors are found, then log them into SUS_SCRATCH_1 bit[0]. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 6 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **High-Speed I/O (HSIO) Configuration** | | | | | | |
| **HSIO Lane Assignment** | | | | | | |
| 21 | + 54h | 19:18 | HSIO Lane 0 Select | 0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [0] and HSIO_TX_D [0] pins are PCIe or not used. |
| 21 | + 54h | 21:20 | HSIO Lane 1 Select | 0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [1] and HSIO_TX_D [1] pins are PCIe or not used. |
| 21 | + 54h | 23:22 | HSIO Lane 2 Select | 0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [2] and HSIO_TX_D [2] pins are PCIe or not used. |
| 21 | + 54h | 25:24 | HSIO Lane 3 Select | 0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [3] and HSIO_TX_D [3] pins are PCIe or not used. |
| 19 | + 4Ch | 9:8 | HSIO Lane 4 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [4] and HSIO_TX_D [4] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 4 to Port 0" to match this selection. |
| 19 | + 4Ch | 11:10 | HSIO Lane 5 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [5] and HSIO_TX_D [5] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 5 to Port 1" to match this selection. |
| 19 | + 4Ch | 13:12 | HSIO Lane 6 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [6] and HSIO_TX_D [6] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 6 to Port 2" to match this selection. |
| 20 | + 50h | 1:0 | HSIO Lane 7 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [7] and HSIO_TX_D [7] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 7 to Port 3" to match this selection. |
| 20 | + 50h | 3:2 | HSIO Lane 8 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane is not Used | Board Dependent | Tells the SoC to select if HSIO_RX_D [8] and HSIO_TX_D [8] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 8 to Port 4" to match this selection. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 7 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 20 | + 50h | 5:4 | HSIO Lane 9 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [9] and HSIO_TX_D [9] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 9 to Port 5" to match this selection. |
| 20 | + 50h | 7:6 | HSIO Lane 10 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [10] and HSIO_TX_D [10] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 10 to Port 6" to match this selection. |
| 20 | + 50h | 9:8 | HSIO Lane 11 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [11] and HSIO_TX_D [11] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 0 - Assign HSIO Lane 11 to Port 7" to match this selection. |
| 20 | + 50h | 11:10 | HSIO Lane 12 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [12] and HSIO_TX_D [12] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 12 to Port 0" to match this selection. |
| 20 | + 50h | 13:12 | HSIO Lane 13 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [13] and HSIO_TX_D [13] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 13 to Port 1" to match this selection. |
| 20 | + 50h | 15:14 | HSIO Lane 14 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [14] and HSIO_TX_D [14] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 14 to Port 2" to match this selection. |
| 21 | + 54h | 17:16 | HSIO Lane 15 Select | 0x0; SATA<br>0x1; PCIe<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [15] and HSIO_TX_D [15] pins are SATA, PCIe or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 15 to Port 3" to match this selection. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 8 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 20 | + 50h | 25:24 | HSIO Lane 16 Select | 0x0; USB 3.0<br>0x1; SATA<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [16] and HSIO_TX_D [16] pins are SATA or USB 3.0 or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 16 to Port 4" to match this selection. |
| 20 | + 50h | 27:26 | HSIO Lane 17 Select | 0x0; USB 3.0<br>0x1; SATA<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [17] and HSIO_TX_D [17] pins are SATA or USB 3.0 or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 17 to Port 5" to match this selection. |
| 20 | + 50h | 29:28 | HSIO Lane 18 Select | 0x0; USB 3.0<br>0x1; SATA<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [18] and HSIO_TX_D [18] pins are SATA or USB 3.0 or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 18 to Port 6" to match this selection. |
| 20 | + 50h | 31:30 | HSIO Lane 19 Select | 0x0; USB 3.0<br>0x1; SATA<br>0x2; HSIO Lane not used | Board Dependent | Tells the SoC to select if HSIO_RX_D [19] and HSIO_TX_D [19] pins are SATA or USB 3.0 or not used.<br>**Note:** Set soft strap "SATA Controller 1 - Assign HSIO Lane 19 to Port 7" to match this selection. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 9 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **HSIO Lane Configuration from SATA Controller** | | | | | | |
| 26 | + 68h | 1:0 | SATA Controller 0 - Assign HSIO Lane 4 to Port 0 | 0x0; HSIO Lane 4 Statically assigned to SATA Controller 0 Port 0<br>0x1; HSIO Lane 4 Not assigned to SATA Controller 0 Port 0 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [4] and HSIO_TX_D [4] pins are assigned to Port 0<br>**Note:** Set the "HSIO Lane 4 Select" soft strap to match this selection |
| 26 | + 68h | 3:2 | SATA Controller 0 - Assign HSIO Lane 5 to Port 1 | 0x0; HSIO Lane 5 Statically assigned to SATA Controller 0 Port 1<br>0x1; HSIO Lane 5 Not assigned to SATA Controller 0 Port 1 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [5] and HSIO_TX_D [5] pins are assigned to Port 1<br>**Note:** Set the "HSIO Lane 5 Select" soft strap to match this selection |
| 26 | + 68h | 5:4 | SATA Controller 0 - Assign HSIO Lane 6 to Port 2 | 0x0; HSIO Lane 6 Statically assigned to SATA Controller 0 Port 2<br>0x1; HSIO Lane 6 Not assigned to SATA Controller 0 Port 2 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [6] and HSIO_TX_D [6] pins are assigned to Port 2<br>**Note:** Set the "HSIO Lane 6 Select" soft strap to match this selection |
| 26 | + 68h | 7:6 | SATA Controller 0 - Assign HSIO Lane 7 to Port 3 | 0x0; HSIO Lane 7 Statically assigned to SATA Controller 0 Port 3<br>0x1; HSIO Lane 7 Not assigned to SATA Controller 0 Port 3 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [7] and HSIO_TX_D [7] pins are assigned to Port 3<br>**Note:** Set the "HSIO Lane 7 Select" soft strap to match this selection |
| 26 | + 68h | 9:8 | SATA Controller 0 - Assign HSIO Lane 8 to Port 4 | 0x0; HSIO Lane 8 Statically assigned to SATA Controller 0 Port 4<br>0x1; HSIO Lane 8 Not assigned to SATA Controller 0 Port 4 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [8] and HSIO_TX_D [8] pins are assigned to Port 4<br>**Note:** Set the "HSIO Lane 8 Select" soft strap to match this selection |
| 26 | + 68h | 11:10 | SATA Controller 0 - Assign HSIO Lane 9 to Port 5 | 0x0; HSIO Lane 9 Statically assigned to SATA Controller 0 Port 5<br>0x1; HSIO Lane 9 Not assigned to SATA Controller 0 Port 5 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [9] and HSIO_TX_D [9] pins are assigned to Port 5<br>**Note:** Set the "HSIO Lane 9 Select" soft strap to match this selection |
| 26 | + 68h | 13:12 | SATA Controller 0 - Assign HSIO Lane 10 to Port 6 | 0x0; HSIO Lane 10 Statically assigned to SATA Controller 0 Port 6<br>0x1; HSIO Lane 10 Not assigned to SATA Controller 0 Port 6 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [10] and HSIO_TX_D [10] pins are assigned to Port 6<br>**Note:** Set the "HSIO Lane 10 Select" soft strap to match this selection |
| 26 | + 68h | 15:14 | SATA Controller 0 - Assign HSIO Lane 11 to Port 7 | 0x0; HSIO Lane 11 Statically assigned to SATA Controller 0 Port 7<br>0x1; HSIO Lane 11 Not assigned to SATA Controller 0 Port 7 | Board Dependent | Tells the SoC SATA Controller 0 if HSIO_RX_D [11] and HSIO_TX_D [11] pins are assigned to Port 7<br>**Note:** Set the "HSIO Lane 11 Select" soft strap to match this selection |
| 27 | + 6Ch | 1:0 | SATA Controller 1 - Assign HSIO Lane 12 to Port 0 | 0x0; HSIO Lane 12 Statically assigned to SATA Controller 1 Port 0<br>0x1; HSIO Lane 12 Not assigned to SATA Controller 1 Port 0 | Board Dependent | Tells the SoC's SATA Controller 1 if HSIO_RX_D [12] and HSIO_TX_D [12] pins are assigned to Port 0<br>**Note:** Set the "HSIO Lane 12 Select" soft strap to match this selection |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 10 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 27 | + 6Ch | 3:2 | SATA Controller 1 - Assign HSIO Lane 13 to Port 1 | 0x0; HSIO Lane 13 Statically assigned to SATA Controller 1 Port 1<br>0x1; HSIO Lane 13 Not assigned to SATA Controller 1 Port 1 | Board Dependent | Tells the SoC's SATA Controller 1 if HSIO_RX_D [13] and HSIO_TX_D [13] pins are assigned to Port 1<br>**Note:** Set the "HSIO Lane 13 Select" soft strap to match this selection |
| 27 | + 6Ch | 5:4 | SATA Controller 1 - Assign HSIO Lane 14 to Port 2 | 0x0; HSIO Lane 14 Statically assigned to SATA Controller 1 Port 2<br>0x1; HSIO Lane 14 Not assigned to SATA Controller 1 Port 2 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [14] and HSIO_TX_D [14] pins are assigned to Port 2<br>**Note:** Set the "HSIO Lane 14 Select" soft strap to match this selection |
| 27 | + 6Ch | 7:6 | SATA Controller 1 - Assign HSIO Lane 15 to Port 3 | 0x0; HSIO Lane 15 Statically assigned to SATA Controller 1 Port 3<br>0x1; HSIO Lane 15 Not assigned to SATA Controller 1 Port 3 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [15] and HSIO_TX_D [15] pins are assigned to Port 3<br>**Note:** Set the "HSIO Lane 15 Select" soft strap to match this selection |
| 27 | + 6Ch | 9:8 | SATA Controller 1 - Assign HSIO Lane 16 to Port 4 | 0x0; HSIO Lane 16 Statically assigned to SATA Controller 1 Port 4<br>0x1; HSIO Lane 16 Not assigned to SATA Controller 1 Port 4 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [16] and HSIO_TX_D [16] pins are assigned to Port 4<br>**Note:** Set the "HSIO Lane 16 Select" soft strap to match this selection |
| 27 | + 6Ch | 11:10 | SATA Controller 1 - Assign HSIO Lane 17 to Port 5 | 0x0; HSIO Lane 17 Statically assigned to SATA Controller 1 Port 5<br>0x1; HSIO Lane 17 Not assigned to SATA Controller 1 Port 5 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [17] and HSIO_TX_D [17] pins are assigned to Port 5<br>**Note:** Set the "HSIO Lane 17 Select" soft strap to match this selection |
| 27 | + 6Ch | 13:12 | SATA Controller 1 - Assign HSIO Lane 18 to Port 6 | 0x0; HSIO Lane 18 Statically assigned to SATA Controller 1 Port 6<br>0x1; HSIO Lane 18 Not assigned to SATA Controller 1 Port 6 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [18] and HSIO_TX_D [18] pins are assigned to Port 6<br>**Note:** Set the "HSIO Lane 18 Select" soft strap to match this selection |
| 27 | + 6Ch | 15:14 | SATA Controller 1 - Assign HSIO Lane 19 to Port 7 | 0x0; HSIO Lane 19 Statically assigned to SATA Controller 1 Port 7<br>0x1; HSIO Lane 19 Not assigned to SATA Controller 1 Port 7 | Board Dependent | Tells the SoC SATA Controller 1 if HSIO_RX_D [19] and HSIO_TX_D [19] pins are assigned to Port 7<br>**Note:** Set the "HSIO Lane 19 Select" soft strap to match this selection |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 11 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **PCIe Bifurcation and Settings** | | | | | | |
| **PCIe Root Port Cluster 0** | | | | | | |
| **PCIe RP Cluster 0 - RP Enable** | | | | | | |
| 24 | + 60h | 0 | PCIe Root Port Cluster 0 - Root Port 0 Enable | 0x0; RP0 Disabled<br>0x1; RP0 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Root Port 0<br>***Note:*** If Root port is set to "0x0 - RP0 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 24 | + 60h | 1 | PCIe Root Port Cluster 0 - Root Port 1 Enable | 0x0; RP1 Disabled<br>0x1; RP1 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Root Port 1<br>***Note:*** If Root port is set to "0x0 - RP1 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 24 | + 60h | 2 | PCIe Root Port Cluster 0 - Root Port 2 Enable | 0x0; RP2 Disabled<br>0x1; RP2 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Root Port 2<br>***Note:*** If Root port is set to "0x0 - RP2 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 24 | + 60h | 3 | PCIe Root Port Cluster 0 - Root Port 3 Enable | 0x0; RP3 Disabled<br>0x1; RP3 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Root Port 3<br>***Note:*** If Root port is set to "0x0 - RP3 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 12 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| | | | | | | **PCIe RP Cluster 0 - RP x1 Link Width Enable** |
| 24 | + 60h | 27:24 | PCIe Root Port Cluster 0<br>- Root Port 0 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 0 Disable" and "PCIe Root Port Cluster 0 - Root Port 0 Enable" are set to Enable, tell the SoC to set RP 0 link width to x1.<br>**Note:** If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 24 | + 60h | 31:28 | PCIe Root Port Cluster 0<br>- Root Port 1 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 0 Disable" and "PCIe Root Port Cluster 0 - Root Port 1 Enable" are set to Enable, tell the SoC to set RP 1 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 25 | + 64h | 3:0 | PCIe Root Port Cluster 0<br>- Root Port 2 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 0 Disable" and "PCIe Root Port Cluster 0 - Root Port 2 Enable" are set to Enable, tell the SoC to set RP 2 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 25 | + 64h | 7:4 | PCIe Root Port Cluster 0<br>- Root Port 3 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 0 Disable" and "PCIe Root Port Cluster 0 - Root Port 3 Enable" are set to Enable, tell the SoC to set RP 3 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 13 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| | | | | | | **PCIe RP Cluster 0 - SRIS Enable** |
| 25 | + 64h | 25 | PCIe Root Port Cluster 0 - Root Port 0 SRIS Enable | 0x0; RP 0 SRIS Disabled<br>0x1; RP 0 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 0. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 25 | + 64h | 26 | PCIe Root Port Cluster 0 - Root Port 1 SRIS Enable | 0x0; RP 1 SRIS Disabled<br>0x1; RP 1 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 1. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 25 | + 64h | 27 | PCIe Root Port Cluster 0 - Root Port 2 SRIS Enable | 0x0; RP 2 SRIS Disabled<br>0x1; RP 2 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 2. SRIS is supported at Gen 1, 2and 2<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 25 | + 64h | 28 | PCIe Root Port Cluster 0 - Root Port 3 SRIS Enable | 0x0; RP 3 SRIS Disabled<br>0x1; RP 3 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 0 Disable" is set to "PCIe Root Port Cluster 0 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 3. SRIS is supported at Gen 1, and 2<br>**Note:** Refer to the SOSCTL.SRISEN register. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 14 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **PCIe Root Port Cluster 1** | | | | | | |
| **PCIe RP Cluster 1 - RP Enable** | | | | | | |
| 22 | + 58h | 0 | PCIe Root Port Cluster 1<br>- Root Port 4 Enable | **0x0;** RP4 Disabled<br>**0x1;** RP4 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Root Port 4.<br>***Note:*** If Root port is set to "0x0 - RP4 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 22 | + 58h | 1 | PCIe Root Port Cluster 1<br>- Root Port 5 Enable | **0x0**; RP5 Disabled<br>**0x1**; RP5 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Root Port 5<br>***Note:*** If Root port is set to "0x0 - RP5 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 22 | + 58h | 2 | PCIe Root Port Cluster 1<br>- Root Port 6 Enable | **0x0;** RP6 Disabled<br>**0x1;** RP6 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Root Port 6<br>***Note:*** If Root port is set to "0x0 - RP6 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |
| 22 | + 58h | 3 | PCIe Root Port Cluster 1<br>- Root Port 7 Enable | **0x0;** RP7 Disabled<br>**0x1;** RP7 Enabled | 0x1 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Root Port 7<br>***Note:*** If Root port is set to "0x0 - RP7 Disabled" then BIOS will not be able to turn it back on. Soft Straps have priority.<br>Bifurcation control register BIFCTL0[2:0] configuration should take into account any disabled root ports via soft straps. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 15 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| | | | | | | **PCIe RP Cluster 1 - RP x1 Link Width Enable** |
| 22 | + 58h | 27:24 | PCIe Root Port Cluster 1<br>- Root Port 4 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 1 Disable" and "PCIe Root Port Cluster 1 - Root Port 4 Enable" are set to Enable, tell the SoC to set RP 4 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 22 | + 58h | 31:28 | PCIe Root Port Cluster 1<br>- Root Port 5 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 1 Disable" and "PCIe Root Port Cluster 1 - Root Port 5 Enable" are set to Enable, tell the SoC to set RP 5 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 23 | + 5Ch | 3:0 | PCIe Root Port Cluster 1<br>- Root Port 6 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 1 Disable" and "PCIe Root Port Cluster 1 - Root Port 6 Enable" are set to Enable, tell the SoC to set RP 6 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |
| 23 | + 5Ch | 7:4 | PCIe Root Port Cluster 1<br>- Root Port 7 x1 Link Width Enable | 0x0; RP Link Width set by BICTRL register<br>0xF; x1 | 0x0 | If soft straps "PCIe Root Port Cluster 1 Disable" and "PCIe Root Port Cluster 1 - Root Port 7 Enable" are set to Enable, tell the SoC to set RP 7 link width to x1.<br>**Notes:**<br>- If setting this soft strap to the "x1" configuration and desiring to use the other HSIO lane for the SATA controller, make sure to set the respective HSIO and SATA controller soft straps accordingly. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 16 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| | | | | | | **PCIe RP Cluster 1 - SRIS Enable** |
| 23 | + 5Ch | 25 | PCIe Root Port Cluster 1 - Root Port 4 SRIS Enable | **0x0;** RP 4 SRIS Disabled<br>**0x1;** RP 4 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 4. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 23 | + 5Ch | 26 | PCIe Root Port Cluster 1 - Root Port 5 SRIS Enable | **0x0;** RP 5 SRIS Disabled<br>**0x1;** RP 5 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 5. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 23 | + 5Ch | 27 | PCIe Root Port Cluster 1 - Root Port 6 SRIS Enable | **0x0;** RP 6 SRIS Disabled<br>**0x1;** RP 6 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 6. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |
| 23 | + 5Ch | 28 | PCIe Root Port Cluster 1 - Root Port 7 SRIS Enable | **0x0;** RP 7 SRIS Disabled<br>**0x1;** RP 7 SRIS Enabled | 0x0 | If soft strap "PCIe Root Port Cluster 1 Disable" is set to "PCIe Root Port Cluster 1 Enabled" tell the SoC to enable Separate Reference Clocks with Independent Spread Spectrum Clocking (SRIS) for Root Port 7. SRIS is supported at Gen 1, and 2.<br>**Note:** Refer to the SOSCTL.SRISEN register. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 17 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **External Clocks** | | | | | | |
| 5 | + 14h | 21 | HSIO Spread Spectrum Clocking Disable | 0x0: Enable HSIO Spread Spectrum Clocking<br>0x1: Disable HSIO Spread Spectrum Clocking | 0x0 | Tells the SoC if HSIO Spread Spectrum Clock (SSC) will be disabled from the time the clock is first output by the SoC for PCIe, SATA, and XHCI.<br>***Note:*** Should BIOS configure this setting, disabling the HSIO SSC will not take effect until after memory training during BIOS execution. |
| 15 | + 3Ch | 2 | DDR Spread Spectrum Clock (SSC) Modulator Disable | 0x0; DDR SSC Disabled<br>0x1; DDR SSC Enabled | 0x1 | Tells the SoC if DDR Spread Spectrum Clock (SSC) modulator will be disabled. |
| **CLK_OUT_D 0 Configuration** | | | | | | |
| 15 | + 3Ch | 8 | Force CLK_OUT_D 0 | 0x0; CLK_OUT_D 0 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps (CLK_OUT_D 0 ; PCIE_CLKREQ_N[x]).<br>0x1; Output Clock to CLK_OUT_D 0 at all times regardless of clkreq setting. | Board Dependent | Tells the SoC to output 100 MHz Clock to CLK_OUT_DP[0] and CLK_OUT_DN[0]<br>***Note:*** There is no option to disable CLK_OUT_D 0. Instead we can prevent a clock from being sent by setting this soft strap to option:<br>"0x0; CLK_OUT_D 0 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps"<br>as well as setting all the CLK_OUT_D 0 -<br>PCIE_CLKREQ_N[0-7] soft straps to: "0x0; Clock will not respond to PCIE_CLKREQ_N[x] pin." |
| 15 | + 3Ch | 16 | CLK_OUT_D 0 - PCIE_CLKREQ_N[0] | 0x0; Clock will not respond to PCIE_CLKREQ_N[0] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[0] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[0] pin. |
| 15 | + 3Ch | 17 | CLK_OUT_D 0 - PCIE_CLKREQ_N[1] | 0x0; Clock will not respond to PCIE_CLKREQ_N[1] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[1] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[1] |
| 15 | + 3Ch | 18 | CLK_OUT_D 0 - PCIE_CLKREQ_N[2] | 0x0; Clock will not respond to PCIE_CLKREQ_N[2] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[2] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[2] |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 18 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 15 | + 3Ch | 19 | CLK_OUT_D 0 - PCIE_CLKREQ_N[3] | 0x0; Clock will not respond to PCIE_CLKREQ_N[3] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[3] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[3] |
| 15 | + 3Ch | 20 | CLK_OUT_D 0 - PCIE_CLKREQ_N[4] | 0x0; Clock will not respond to PCIE_CLKREQ_N[4] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[4] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[4] |
| 15 | + 3Ch | 21 | CLK_OUT_D 0 - PCIE_CLKREQ_N[5] | 0x0; Clock will not respond to PCIE_CLKREQ_N[5] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[5] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[5] |
| 15 | + 3Ch | 22 | CLK_OUT_D 0 - PCIE_CLKREQ_N[6] | 0x0; Clock will not respond to PCIE_CLKREQ_N[6] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[6] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[6] |
| 15 | + 3Ch | 23 | CLK_OUT_D 0 - PCIE_CLKREQ_N[7] | 0x0; Clock will not respond to PCIE_CLKREQ_N[7] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[7] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 0" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[0] and CLK_OUT_DN[0] will respond to PCIE_CLKREQ_N[7] |
| **CLK_OUT_D 1 Configuration** | | | | | | |
| 15 | + 3Ch | 9 | Force CLK_OUT_D 1 | 0x0; CLK_OUT_D 1 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps (CLK_OUT_D 1 ; PCIE_CLKREQ_N[x]).<br>0x1; Output Clock to CLK_OUT_D 1 at all times regardless of clkreq setting. | Board Dependent | Tells the SoC to output 100 MHz Clock to CLK_OUT_DP[1] and CLK_OUT_DN[1]<br>***Note:*** There is no option to disable CLK_OUT_D 1. Instead we can prevent a clock from being sent by setting this soft strap to option:<br>"0x0; CLK_OUT_D 1 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps"<br>as well as setting all the CLK_OUT_D 1 -<br>PCIE_CLKREQ_N[0-7] soft straps to: "0x0; Clock will not respond to PCIE_CLKREQ_N[x] pin." |
| 15 | + 3Ch | 24 | CLK_OUT_D 1 - PCIE_CLKREQ_N[0] | 0x0; Clock will not respond to PCIE_CLKREQ_N[0] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[0] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[0] pin. |
| 15 | + 3Ch | 25 | CLK_OUT_D 1 - PCIE_CLKREQ_N[1] | 0x0; Clock will not respond to PCIE_CLKREQ_N[1] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[1] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[1] |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 19 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 15 | + 3Ch | 26 | CLK_OUT_D 1 - PCIE_CLKREQ_N[2] | 0x0; Clock will not respond to PCIE_CLKREQ_N[2] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[2] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[2] |
| 15 | + 3Ch | 27 | CLK_OUT_D 1 - PCIE_CLKREQ_N[3] | 0x0; Clock will not respond to PCIE_CLKREQ_N[3] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[3] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[3] |
| 15 | + 3Ch | 28 | CLK_OUT_D 1 - PCIE_CLKREQ_N[4] | 0x0; Clock will not respond to PCIE_CLKREQ_N[4] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[4] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[4] |
| 15 | + 3Ch | 29 | CLK_OUT_D 1 - PCIE_CLKREQ_N[5] | 0x0; Clock will not respond to PCIE_CLKREQ_N[5] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[5] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[5] |
| 15 | + 3Ch | 30 | CLK_OUT_D 1 - PCIE_CLKREQ_N[6] | 0x0; Clock will not respond to PCIE_CLKREQ_N[6] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[6] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[6] |
| 15 | + 3Ch | 31 | CLK_OUT_D 1 - PCIE_CLKREQ_N[7] | 0x0; Clock will not respond to PCIE_CLKREQ_N[7] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[7] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 1" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[1] and CLK_OUT_DN[1] will respond to PCIE_CLKREQ_N[7] |
| **CLK_OUT_D 2 Configuration** | | | | | | |
| 15 | + 3Ch | 10 | Force CLK_OUT_D 2 | 0x0; CLK_OUT_D 2 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps (CLK_OUT_D 2 ; PCIE_CLKREQ_N[x]).<br>0x1; Output Clock to CLK_OUT_D 2 at all times regardless of clkreq setting. | Board Dependent | Tells the SoC to output 100 MHz Clock to CLK_OUT_DP[2] and CLK_OUT_DN[2]<br>***Note:*** There is no option to disable CLK_OUT_D 2. Instead we can prevent a clock from being sent by setting this soft strap to option:<br>"0x0; CLK_OUT_D 2 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps"<br>as well as setting all the CLK_OUT_D 2 -<br>PCIE_CLKREQ_N[0-7] soft straps to: "0x0; Clock will not respond to PCIE_CLKREQ_N[x] pin." |
| 16 | + 40h | 0 | CLK_OUT_D 2 - PCIE_CLKREQ_N[0] | 0x0; Clock will not respond to PCIE_CLKREQ_N[0] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[0] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[0] pin. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 20 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 16 | + 40h | 1 | CLK_OUT_D 2 - PCIE_CLKREQ_N[1] | 0x0; Clock will not respond to PCIE_CLKREQ_N[1] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[1] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[1] |
| 16 | + 40h | 2 | CLK_OUT_D 2 - PCIE_CLKREQ_N[2] | 0x0; Clock will not respond to PCIE_CLKREQ_N[2] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[2] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[2] |
| 16 | + 40h | 3 | CLK_OUT_D 2 - PCIE_CLKREQ_N[3] | 0x0; Clock will not respond to PCIE_CLKREQ_N[3] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[3] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[3] |
| 16 | + 40h | 4 | CLK_OUT_D 2 - PCIE_CLKREQ_N[4] | 0x0; Clock will not respond to PCIE_CLKREQ_N[4] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[4] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[4] |
| 16 | + 40h | 5 | CLK_OUT_D 2 - PCIE_CLKREQ_N[5] | 0x0; Clock will not respond to PCIE_CLKREQ_N[5] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[5] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[5] |
| 16 | + 40h | 6 | CLK_OUT_D 2 - PCIE_CLKREQ_N[6] | 0x0; Clock will not respond to PCIE_CLKREQ_N[6] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[6] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[6] |
| 16 | + 40h | 7 | CLK_OUT_D 2 - PCIE_CLKREQ_N[7] | 0x0; Clock will not respond to PCIE_CLKREQ_N[7] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[7] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 2" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[2] and CLK_OUT_DN[2] will respond to PCIE_CLKREQ_N[7] |
| **CLK_OUT_D 3 Configuration** | | | | | | |
| 15 | + 3Ch | 11 | Force CLK_OUT_D 3 | 0x0; CLK_OUT_D 3 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps (CLK_OUT_D 3 ; PCIE_CLKREQ_N[x]).<br>0x1; Output Clock to CLK_OUT_D 3 at all times regardless of clkreq setting. | Board Dependent | Tells the SoC to output 100 MHz Clock to CLK_OUT_DP[3] and CLK_OUT_DN[3]<br>***Note:*** There is no option to disable CLK_OUT_D 3. Instead we can prevent a clock from being sent by setting this soft strap to option:<br>"0x0; CLK_OUT_D 3 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps"<br>as well as setting all the CLK_OUT_D 3 -<br>PCIE_CLKREQ_N[0-7] soft straps to: "0x0; Clock will not respond to PCIE_CLKREQ_N[x] pin." |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 21 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 16 | + 40h | 8 | CLK_OUT_D 3 - PCIE_CLKREQ_N[0] | 0x0; Clock will not respond to PCIE_CLKREQ_N[0] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[0] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[0] pin. |
| 16 | + 40h | 9 | CLK_OUT_D 3 - PCIE_CLKREQ_N[1] | 0x0; Clock will not respond to PCIE_CLKREQ_N[1] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[1] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[1] |
| 16 | + 40h | 10 | CLK_OUT_D 3 - PCIE_CLKREQ_N[2] | 0x0; Clock will not respond to PCIE_CLKREQ_N[2] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[2] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[2] |
| 16 | + 40h | 11 | CLK_OUT_D 3 - PCIE_CLKREQ_N[3] | 0x0; Clock will not respond to PCIE_CLKREQ_N[3] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[3] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[3] |
| 16 | + 40h | 12 | CLK_OUT_D 3 - PCIE_CLKREQ_N[4] | 0x0; Clock will not respond to PCIE_CLKREQ_N[4] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[4] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[4] |
| 16 | + 40h | 13 | CLK_OUT_D 3 - PCIE_CLKREQ_N[5] | 0x0; Clock will not respond to PCIE_CLKREQ_N[5] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[5] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[5] |
| 16 | + 40h | 14 | CLK_OUT_D 3 - PCIE_CLKREQ_N[6] | 0x0; Clock will not respond to PCIE_CLKREQ_N[6] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[6] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[6] |
| 16 | + 40h | 15 | CLK_OUT_D 3 - PCIE_CLKREQ_N[7] | 0x0; Clock will not respond to PCIE_CLKREQ_N[7] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[7] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 3" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[3] and CLK_OUT_DN[3] will respond to PCIE_CLKREQ_N[7] |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 22 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **CLK_OUT_D 4 Configuration** | | | | | | |
| 15 | + 3Ch | 12 | Force CLK_OUT_D 4 | 0x0; CLK_OUT_D 4 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps (CLK_OUT_D 4 ; PCIE_CLKREQ_N[x]).<br>0x1; Output Clock to CLK_OUT_D 4 at all times regardless of clkreq setting. | Board Dependent | Tells the SoC to output 100 MHz Clock to CLK_OUT_DP[4] and CLK_OUT_DN[4]<br>***Note:*** There is no option to disable CLK_OUT_D 4. Instead we can prevent a clock from being sent by setting this soft strap to option:<br>"0x0; CLK_OUT_D 4 will respond to PCIE_CLKREQ_N[x] pins configured through soft straps"<br>as well as setting all the CLK_OUT_D 4 -<br>PCIE_CLKREQ_N[0-7] soft straps to: "0x0; Clock will not respond to PCIE_CLKREQ_N[x] pin." |
| 16 | + 40h | 16 | CLK_OUT_D 4 - PCIE_CLKREQ_N[0] | 0x0; Clock will not respond to PCIE_CLKREQ_N[0] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[0] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[0] pin. |
| 16 | + 40h | 17 | CLK_OUT_D 4 - PCIE_CLKREQ_N[1] | 0x0; Clock will not respond to PCIE_CLKREQ_N[1] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[1] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[1] |
| 16 | + 40h | 18 | CLK_OUT_D 4 - PCIE_CLKREQ_N[2] | 0x0; Clock will not respond to PCIE_CLKREQ_N[2] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[2] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[2] |
| 16 | + 40h | 19 | CLK_OUT_D 4 - PCIE_CLKREQ_N[3] | 0x0; Clock will not respond to PCIE_CLKREQ_N[3] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[3] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[3] |
| 16 | + 40h | 20 | CLK_OUT_D 4 - PCIE_CLKREQ_N[4] | 0x0; Clock will not respond to PCIE_CLKREQ_N[4] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[4] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[4] |
| 16 | + 40h | 21 | CLK_OUT_D 4 - PCIE_CLKREQ_N[5] | 0x0; Clock will not respond to PCIE_CLKREQ_N[5] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[5] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[5] |
| 16 | + 40h | 22 | CLK_OUT_D 4 - PCIE_CLKREQ_N[6] | 0x0; Clock will not respond to PCIE_CLKREQ_N[6] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[6] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[6] |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 23 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 16 | + 40h | 23 | CLK_OUT_D 4 - PCIE_CLKREQ_N[7] | 0x0; Clock will not respond to PCIE_CLKREQ_N[7] pin.<br>0x1; Clock will respond to PCIE_CLKREQ_N[7] pin. | Board Dependent | If soft strap "Force CLK_OUT_D 4" is set to respond to PCIE_CLKREQ_N[x]" then tell the SoC if CLK_OUT_DP[4] and CLK_OUT_DN[4] will respond to PCIE_CLKREQ_N[7] |
| **Flex Clocks Configuration** | | | | | | |
| 16 | + 40h | 29:24 | FLEX_CLK_SE[0] - Frequency | 0x04; 50 MHz<br>0x06; 33.3 MHz<br>0x08; 25 MHz<br>0x22; 48 MHz | 0x22 | Tells the SoC what frequency to output from FLEX_CLK_SE[0]. |
| 16 | + 40h | 30 | FLEX_CLK_SE[0] - Isolation Enable | 0x1; Wait to drive the clocks to ground until after DDR Domains are powered on<br>0x0; Drive clocks by default (shortly after RSMRST_N) | 0x0 | This soft strap can help prevent driving a clock into an unpowered domain. |
| 17 | + 44h | 5:0 | FLEX_CLK_SE[1] - Frequency | 0x04; 50 MHz<br>0x06; 33.3 MHz<br>0x08; 25 MHz<br>0x22; 48 MHz | 0x22 | Tells the SoC what frequency to output from FLEX_CLK_SE[1]. |
| 17 | + 44h | 6 | FLEX_CLK_SE[1] - Isolation Enable | 0x1; Wait to drive the clocks to ground until after DDR Domains are powered on<br>0x0; Drive clocks by default (shortly after RSMRST_N) | 0x0 | Isolates FlexIO Clock 1 when host_powergood_rst_b is active to prevent driving clock into an unpowered domain. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 24 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **SPI** | | | | | | |
| 9 | + 24h | 0 | SPI - Dual Output Fast Read Enable (DORE) | 0x0; Dual Output Read is disabled<br>0x1; Dual Output Read is enabled | 0x1 | Tells the SoC SPI controller if Dual Output Fast Read (DORE) instructions are supported. This will only take effect if the SPI flash device connected to the SPI controller also supports DORE in its SFDP table and has the Dual Output bit and the Fast Read Support bit enabled in the FLCOMP register inside the SPI flash descriptor region.<br>***Note:*** Only one soft strap between DORE/DIORE/QORE/QIORE SPI modes can be enabled at a time. i.e. if DORE is enabled via soft strap, then DIORE/QORE/QIORE soft straps must be set to disabled via soft strap. |
| 9 | + 24h | 1 | SPI - Dual I/O Fast Read Enable (DIORE) | 0x0; Dual I/O Read is disabled<br>0x1; Dual I/O Read is enabled | 0x1 | Tells the SoC SPI controller if Dual I/O Fast Read (DIORE) instructions are supported. This will only take effect if the SPI flash device connected to the SPI controller also supports DIORE in its SFDP table and has the Dual Output bit and the Fast Read Support bit enabled in the FLCOMP register inside the SPI flash descriptor region.<br>***Note:*** Only one soft strap between DORE/DIORE/QORE/QIORE SPI modes can be enabled at a time. i.e. if DORE is enabled via soft strap, then DIORE/QORE/QIORE soft straps must be set to disabled via soft strap. |
| 9 | + 24h | 2 | SPI - Quad Output Fast Read Enable (QORE) | 0x0; Quad Output Read is disabled<br>0x1; Quad Output Read is enabled | 0x1 | Tells the SoC SPI controller if Quad Output Fast Read (QORE) instructions are supported. This will only take effect if the SPI flash device connected to the SPI controller also supports QORE in its SFDP table and has the Dual Output bit and the Fast Read Support bit enabled in the FLCOMP register inside the SPI flash descriptor region.<br>***Note:*** Only one soft strap between DORE/DIORE/QORE/QIORE SPI modes can be enabled at a time. i.e. if DORE is enabled via soft strap, then DIORE/QORE/QIORE soft straps must be set to disabled via soft strap. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 25 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 9 | + 24h | 3 | SPI - Quad I/O Fast Read Enable (QIORE) | 0x0; Quad I/O Read is disabled<br>0x1; Quad I/O Read is enabled | 0x1 | Tells the SoC SPI controller if Quad I/O Fast Read (QIORE) instructions are supported. This will only take effect if the SPI flash device connected to the SPI controller also supports QIORE in its SFDP table and has the Dual Output bit and the Fast Read Support bit enabled in the FLCOMP register inside the SPI flash descriptor region.<br>***Note:*** Only one soft strap between DORE/DIORE/QORE/QIORE SPI modes can be enabled at a time. i.e. if DORE is enabled via soft strap, then DIORE/QORE/QIORE soft straps must be set to disabled via soft strap. |
| 9 | + 24h | 6:4 | SPI - Boot Block Size | 0x0; 64KB; Invert A16 if Top Swap is enabled<br>0x1; 128KB; Invert A17 if Top Swap is enabled<br>0x2; 256KB; Invert A18 if Top Swap is enabled<br>0x3; 512KB; Invert A19 if Top Swap is enabled<br>0x4; 1MB; Invert A20 if Top Swap is enabled | 0x0 | Tells the SoC if the size of the Top Swap boot block when booting from the SPI interface.<br>**Note**: If boot interface is LPC, this strap has not effect and the Boot block size is 64 KB |
| 9 | + 24h | 8 | SPI - Enable Delay before write busy poll | 0x1; SPI controller may delay the start of issuing read_status to poll for flash device busy after a write operation (enabled)<br>0x0; SPI controller must start polling immediately after issuing the write command (disabled) | 0x0 | Tells the SoC if SPI delay before write busy poll is enabled |
| 9 | + 24h | 9 | SPI - Enable Delay before erase busy poll | 0x1; SPI controller may delay the start of issuing read_status to poll for flash device busy after an erase operation (enabled)<br>0x0; SPI controller must start polling immediately after issuing the erase command. (disabled) | 0x0 | Tells the SoC if SPI delay before erase busy poll is enabled |
| 9 | + 24h | 10 | SPI - Enable Delay before RPMC busy poll | 0x1; SPI controller may delay the start of polling for flash device busy after an RPMC operation (enabled)<br>0x0; SPI controller must start polling immediately after issuing any RPMC command. (disabled) | 0x0 | Tells the SoC if SPI delay before RPMC busy poll is enabled |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 26 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 9 | + 24h | 11 | SPI - Enable device 0 deep power down | 0x1; flash controller implements enter/exit deep power down to this device if it discovers capability via SFDP (enabled)<br>0x0; flash controller does not implement enter/exit deep power down for this device (disabled) | 0x0 | Tells the SoC to implement SPI device 0 (Flash) deep power down |
| 9 | + 24h | 12 | SPI - Enable device 1 deep power down | 0x1; flash controller implements enter/exit deep power down to this device if it discovers capability via SFDP (enabled)<br>0x0; flash controller does not implement enter/exit deep power down for this device (disabled) | 0x0 | Tells the SoC to implement SPI device 1 (Flash) deep power down |
| 9 | + 24h | 13 | SPI - Host Software Sequencing Enable | 0x0 - Host SW sequencing is disabled. Default to HW sequencing.<br>0x1 - Host SW sequencing is enabled along with HW sequencing. | 0x0 | Tells the SoC that SPI SW sequencing is supported in addition to HW sequencing.<br>This soft strap takes precedence over BIOS_DLOCK.SSEQLOCKDN and BIOS_HSFSTS_CTL.FLOCKDN registers. Should it be set to 0x0, these registers become RO and cannot enable SPI SW sequencing.<br>***Note:*** SPI SW Sequencing is intended as a backup to HW sequencing. |
| 9 | + 24h | 27:25 | SPI - Resume Hold off Delay (tRHD) | 0x0; 0us<br>0x1; 2us<br>0x2; 4us<br>0x3; 6us<br>0x4; 8us<br>0x5; 10us<br>0x6; 12us<br>0x7; 14us | 0x4 | Tells the SoC to specify the time after the completion of a pri_op before the flash controller sends the resume instruction. If a new pri_op is eligible to be issued prior to the end of this delay time then the pri_op is issued and the timer is re-initialized to tRHD. |
| 9 | + 24h | 31:30 | SPI - Resume to suspend Timing | 0x0; 128 us<br>0x1; 256 us<br>0x2; 512 us<br>0x3; no ceiling, use the SFDP values | 0x3 | Tells the SoC the maximum value for the write and erase resume to suspend intervals |
| 10 | + 28h | 10:8 | SPI - TPM Clock Frequency | 0x2; 48 MHz<br>0x4; 30 MHz<br>0x6; 17 MHz | 0x6 | Tells the SoC what clock frequency to use when talking to TPM on SPI<br>***Note:*** To control the speed of the instructions between the SoC SPI controller and the SPI flash memory boot device connected to it, configure the FLCOMP register inside the SPI flash descriptor region. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 27 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 10 | + 28h | 14 | SPI - Ignore Fixed FLMSTR Permission | 0x0; FLMSTR fixed permission to regions is enabled<br>0x1; FLMSTR fixed permission to regions may be disabled | 0x0 | Tells the SoC to ignore Fixed FLMSTR Permission settings that are set in the FLMSTR registers inside the descriptor region of the SPI flash memory boot device connected to the SoC to force the SPI controller to ignore each Master's default permission to its own region(s).<br>This affects FLMSTR default permissions and a software master's capability to extend permission to its region(s). |
| 11 | + 2Ch | 31:0 | SPI - Global Protected Range default (GPRD) | Value: Hex | 0x00000000 | Tells the SoC to set the default value of the GPR0 register in the SPI Flash Controller as a security feature. This soft strap value is intended to be configured by customer. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 28 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **PECI** | | | | | | |
| 0 | + 0h | 6:0 | PECI - Target Write Address | Value: Hex | 0x4C | The SoC reads this strap and programs the ADDR0 field of the Target Address Control Register (TACTRL) located at sideband Port CEh, offset 228h. |
| 1 | + 4h | 26:20 | PECI - Target Read Address | Value: Hex | 0x4B | The SoC reads this strap and programs the GPTRADR field of the General Control Register (GPBRCTRL) located at sideband Port CEh, offset 240h |
| **Top Swap Override** | | | | | | |
| 30 | + 78h | 1 | Protected Range Top Swap Override Enable | 0x0; PRR_TS_OVR is Read Only (disabled)<br>0x1; Allows PRR_TS_OVR register to be Read/Write. (enabled) | 0x0 | Tells the SoC if Protected Range Top Swap Override is enabled by setting the permission to register **PRR_TS_OVR** to Read/Write or Read Only. |
| **TPM over SPI or LPC** | | | | | | |
| 30 | + 78h | 0 | TPM Over SPI or LPC Select | 0x0 - TPM on LPC<br>0x1 -TPM on SPI | 0x0 | Tells the SoC to enable TPM on SPI or LPC interfaces. The Intel Manageability Engine (ME) consumes this soft strap. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 29 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Intel® Management Engine (Intel® ME)** | | | | | | |
| 14 | + 38h | 1 | ME ROM - Intel® Trace Hub Soft Enable | 0x0; Tracing is disabled<br>0x1; Tracing is enabled | 0x0 | This soft strap enables Intel Trace Hub-based tracing in ME ROM.<br>**Note:** In order to use Intel Trace Hub over USB3.0, SS14 bit[17] - "ME - Intel® Direct Connect Interface Enabled" must be set to 0x1 - Enabled. |
| 14 | + 38h | 17 | ME - Intel® Direct Connect Interface Enabled | 0x0; Intel® DCI Disabled<br>0x1; Intel® DCI Enabled | 0x0 | Tells the SoC to enable Intel® Direct Connect Interface (Intel® DCI).<br>**Note:** This is required to enable debugging over USB3.0 interface.<br>This soft strap is also required for Intel Trace Hub to work over USB3.0<br>For more information on Intel DCI. See EDS Chapter 38.4 |
| 14 | + 38h | 21 | ME - Intel® Trace Hub - Emergency Mode | 0x0; ROM Tracing Emergency mode disabled<br>0x1; ROM Tracing Emergency mode enabled | 0x0 | Tells the SoC to enable ROM Tracing in the base platform ME image.<br>This is for DEBUG ONLY should not be set for production environment<br>**Note:** Enabling this soft strap will increase the ME loading time due to ME outputting Debug traces and thus increase overall system boot time. |
| **Intel® ME SMBus Management Transport 0 - SMT 0 Configuration** | | | | | | |
| 31 | + 7Ch | 8 | ME SMBus Management Transport 1 - SMT 0 SMBus Enable | 0x0; SMT disabled<br>0x1; SMT enabled | 0x1 | Tells the SoC that SMT0 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 35 | + 8Ch | 17:16 | ME SMBus Management Transport (SMT) 0 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 33 | + 84h | 8 | ME SMBus Management Transport (SMT) 0 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x0 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to "1" when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 30 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 31 | + 7Ch | 16 | ME SMBus Management Transport (SMT) 0 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 31 | + 7Ch | 23:17 | ME SMBus Management Transport (SMT) 0 - General Purpose (GP) Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Note:** This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 32 | + 80h | 16 | ME SMBus Management Transport (SMT) 0 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit) |
| 31 | + 7Ch | 30:24 | ME SMBus Management Transport (SMT) 0 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>$I^2C$ address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 32 | + 80h | 24 | ME SMBus Management Transport (SMT) 0 - Alert Sending Device (ASD) Address Enable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 32 | + 80h | 6:0 | ME SMBus Management Transport (SMT) 0 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 31 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 33 | + 84h | 0 | ME SMBus Management Transport (SMT) 0 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 32 | + 80h | 14:8 | ME SMBus Management Transport (SMT) 0 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |
| 33 | + 84h | 31:16 | ME SMBus Management Transport (SMT) 0 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 34 | + 88h | 15:0 | ME SMBus Management Transport (SMT) 0 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 34 | + 88h | 31:16 | ME SMBus Management Transport (SMT) 0 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 35 | + 8Ch | 15:0 | ME SMBus Management Transport (SMT) 0 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 32 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Intel® ME SMBus Management Transport 1 - SMT 1 Configuration** | | | | | | |
| 36 | + 90h | 8 | ME SMBus Management Transport (SMT) 1 - SMT 1 SMBus Enable | 0x0; SMT disabled<br>0x1; SMT enabled | 0x1 | Tells the SoC that SMT1 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 40 | + A0h | 17:16 | ME SMBus Management Transport (SMT) 1 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 38 | + 98h | 8 | ME SMBus Management Transport (SMT) 1 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x0 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to "1" when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |
| 36 | + 90h | 16 | ME SMBus Management Transport (SMT) 1 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 36 | + 90h | 23:17 | ME SMBus Management Transport (SMT) 1 - General Purpose (GP) Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Note:** This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 37 | + 94h | 16 | ME SMBus Management Transport (SMT) 1 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit). |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 33 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 36 | + 90h | 30:24 | ME SMBus Management Transport (SMT) 1 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>$I^2C$ address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 37 | + 94h | 24 | ME SMBus Management Transport (SMT) 1 - Alert Sending Device (ASD) Address Enable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 37 | + 94h | 6:0 | ME SMBus Management Transport (SMT) 1 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |
| 38 | + 98h | 0 | ME SMBus Management Transport (SMT) 1 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 37 | + 94h | 14:8 | ME SMBus Management Transport (SMT) 1 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |
| 38 | + 98h | 31:16 | ME SMBus Management Transport (SMT) 1 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 34 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 39 | + 9Ch | 15:0 | ME SMBus Management Transport (SMT) 1 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 39 | + 9Ch | 31:16 | ME SMBus Management Transport (SMT) 1 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 40 | + A0h | 15:0 | ME SMBus Management Transport (SMT) 1 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 35 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Intel® ME SMBus Management Transport 2 - SMT 2 Configuration** | | | | | | |
| 41 | + A4h | 8 | ME SMBus Management Transport (SMT) 2 - SMT 2 SMBus Enable | 0x0; SMT disabled<br>0x1; SMT enabled | 0x1 | Tells the SoC that SMT2 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 45 | + B4h | 17:16 | ME SMBus Management Transport (SMT) 2 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 43 | + ACh | 8 | ME SMBus Management Transport (SMT) 2 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x0 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to "1" when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |
| 41 | + A4h | 16 | ME SMBus Management Transport (SMT) 2 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 41 | + A4h | 23:17 | ME SMBus Management Transport (SMT) 2 - General Purpose (GP) Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 42 | + A8h | 16 | ME SMBus Management Transport (SMT) 2 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit) |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 36 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 41 | + A4h | 30:24 | ME SMBus Management Transport (SMT) 2 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>- address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 42 | + A8h | 24 | ME SMBus Management Transport (SMT) 2 - Alert Sending Device (ASD) Address Enable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 42 | + A8h | 6:0 | ME SMBus Management Transport (SMT) 2 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |
| 43 | + ACh | 0 | ME SMBus Management Transport (SMT) 2 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 42 | + A8h | 14:8 | ME SMBus Management Transport (SMT) 2 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |
| 43 | + ACh | 31:16 | ME SMBus Management Transport (SMT) 2 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 37 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 44 | + B0h | 15:0 | ME SMBus Management Transport (SMT) 2 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 44 | + B0h | 31:16 | ME SMBus Management Transport (SMT) 2 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 45 | + B4h | 15:0 | ME SMBus Management Transport (SMT) 2 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 38 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Innovation Engine (IE)** | | | | | | |
| 47 | + BCh | 13 | IE Disable | 0x0; IE is enabled<br>0x1; IE is disabled | 0x0 | Tells the SoC to Disable the Innovation Engine (IE) |
| 47 | + BCh | 14 | IE ROM - SMT Log Enable | 0x0; ROM log to SMT Disabled<br>0x1; ROM log to SMT Enabled | 0x0 | Tells the SoC to Enable the IE ROM log to SMT0<br>**Note:** IE will run if IE Verified Boot is enabled even if this SS disables the IE. |
| **IE SMBus Management Transport 0 - SMT 0 Configuration** | | | | | | |
| 48 | + C0h | 8 | IE SMBus Management Transport (SMT) 0 - SMT 0 SMBus Enable | 0x0; SMT0 disabled<br>0x1; SMT0 enabled | 0x1 | Tells the SoC that SMT0 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 52 | + D0h | 17:16 | IE SMBus Management Transport (SMT) 0 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 50 | + C8h | 8 | IE SMBus Management Transport (SMT) 0 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x0 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to "1" when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |
| 48 | + C0h | 16 | IE SMBus Management Transport (SMT) 0 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 48 | + C0h | 23:17 | IE SMBus Management Transport (SMT) 0 - General Purpose (GP) Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Note:** This address has to be configured so that it does not conflict with other devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 39 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 49 | + C4h | 16 | IE SMBus Management Transport (SMT) 0 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit) |
| 48 | + C0h | 30:24 | IE SMBus Management Transport (SMT) 0 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>-$I^2C$ address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 49 | + C4h | 24 | IE SMBus Management Transport (SMT) 0 - Alert Sending Device (ASD) Address Enable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 49 | + C4h | 6:0 | IE SMBus Management Transport (SMT) 0 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |
| 50 | + C8h | 0 | IE SMBus Management Transport (SMT) 0 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 49 | + C4h | 14:8 | IE SMBus Management Transport (SMT) 0 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 40 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 50 | + C8h | 31:16 | IE SMBus Management Transport (SMT) 0 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 51 | + CCh | 15:0 | IE SMBus Management Transport (SMT) 0 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 51 | + CCh | 31:16 | IE SMBus Management Transport (SMT) 0 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 52 | + D0h | 15:0 | IE SMBus Management Transport (SMT) 0 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Note:** To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 41 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **IE SMBus Management Transport 1 - SMT 1 Configuration** | | | | | | |
| 53 | + D4h | 8 | IE SMBus Management Transport (SMT) 1 - SMT 1 SMBus Enable | 0x0; SMT1 disabled<br>0x1; SMT1 enabled | 0x1 | Tells the SoC that SMT1 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 57 | + E4h | 17:16 | IE SMBus Management Transport (SMT) 1 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 55 | + DCh | 8 | IE SMBus Management Transport (SMT) 1 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x0 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to **"1"** when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |
| 53 | + D4h | 16 | IE SMBus Management Transport (SMT) 1 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 53 | + D4h | 23:17 | IE SMBus Management Transport (SMT) 1 - GP Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Note:** This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 54 | + D8h | 16 | IE SMBus Management Transport (SMT) 1 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit) |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 42 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 53 | + D4h | 30:24 | IE SMBus Management Transport (SMT) 1 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>-$I^2C$ address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 54 | + D8h | 24 | IE SMBus Management Transport (SMT) 1 - Alert Sending Device (ASD) AddressEnable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 54 | + D8h | 6:0 | IE SMBus Management Transport (SMT) 1 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |
| 55 | + DCh | 0 | IE SMBus Management Transport (SMT) 1 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 54 | + D8h | 14:8 | IE SMBus Management Transport (SMT) 1 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |
| 55 | + DCh | 31:16 | IE SMBus Management Transport (SMT) 1 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Notes:**<br>-To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 43 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 56 | + E0h | 15:0 | IE SMBus Management Transport (SMT) 1 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Notes:**<br>-To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 56 | + E0h | 31:16 | IE SMBus Management Transport (SMT) 1 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Notes:**<br>-To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 57 | + E4h | 15:0 | IE SMBus Management Transport (SMT) 1 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Notes:**<br>-To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 44 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **IE SMBus Management Transport 2 - SMT 2 Configuration** | | | | | | |
| 58 | + E8h | 8 | IE SMBus Management Transport (SMT) 2 - SMT 2 SMBus Enable | 0x0; SMT2 disabled<br>0x1; SMT2 enabled | 0x1 | Tells the SoC that SMT2 is enabled and connected to the SMBus. Disabled means that the device is not connected to the SMBus Pins. |
| 62 | + F8h | 17:16 | IE SMBus Management Transport (SMT) 2 - SMBus Clock | 0x1; Standard (100 KHz)<br>0x2; Fast Mode (400 KHz)<br>0x3; Fast Mode Plus (1 MHz) | 0x1 | Tells the SoC the speed the physical bus must operate at.<br>**Note:** It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed. |
| 60 | + F0h | 8 | IE SMBus Management Transport (SMT) 2 - Host SMBus Address Enable | 0x0; Host SMBus Target Address disabled<br>0x1; Host SMBus Target Address enabled | 0x00 | Tells the SoC SMT controller that it is the SMBus segment owner and any device sending a request to Address 10h should be claimed by the SMT controller.<br>When set, enables HW to accept write cycles targeting address 10h and enables target logic to ACK Host SMBus address (10h) on the SMBus.<br>**Note:** This bit must be set to "1" when HW is configured to support MCTP and chipset is the SMBus Segment Owner. |
| 58 | + E8h | 16 | IE SMBus Management Transport (SMT) 2 - General Purpose (GP) Target Address Enable | 0x0; GP Target Address is Disabled<br>0x1; GP Target Address is Enabled | 0x0 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands. (Address Enable Bit)<br>**Note:** The 'Enable' for this address is programmed through this soft strap. HW does not take any action based on the value of this soft strap. It is simply provided to FW for reading. |
| 58 | + E8h | 23:17 | IE SMBus Management Transport (SMT) 2 - General Purpose (GP) Target Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate with externally initiated Block Read commands.<br>**Note:** This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 59 | + ECh | 16 | IE SMBus Management Transport (SMT) 2 - $I^2C$ Address Enable | 0x0; $I^2C$ Target Address disabled<br>0x1; $I^2C$ Target Address enabled | 0x00 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$ (Address Enable Bit) |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 45 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 58 | + E8h | 30:24 | IE SMBus Management Transport (SMT) 2 - $I^2C$ Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave over $I^2C$.<br>**Notes:**<br>-$I^2C$ address of the SMT logic for support of $I^2C$ target. All $I^2C$ protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.I2C_EN register is set.<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network. |
| 59 | + ECh | 24 | IE SMBus Management Transport (SMT) 2 - Alert Sending Device (ASD) AddressEnable | 0x0; ASD Target Address disabled<br>0x1; ASD Target Address enabled | 0x00 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol. (Address Enable Bit) |
| 59 | + ECh | 6:0 | IE SMBus Management Transport (SMT) 2 - Alert Sending Device (ASD) Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using ASD protocol.<br>**Notes:**<br>-This address has to be configured so that it does not conflict with other devices on the SMBus network.<br>-SMBus address of the SMT logic for support of ASF protocol. All ASF protocol transactions targeting this address will be ACKed by the HW if the TPOLICY.ASD_EN register is set. |
| 60 | + F0h | 0 | IE SMBus Management Transport (SMT) 2 - MCTP Address Enable | 0x0; MCTP Target Address disabled<br>0x1; MCTP Target Address enabled | 0x00 | Tells the SoC to enable the target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol. (Address Enable Bit) |
| 59 | + ECh | 14:8 | IE SMBus Management Transport (SMT) 2 - MCTP Address | Value: Hex | 0x00 | Tells the SoC the 7-bit configurable target slave address of the SMT Controller so that it can communicate as a slave using MCTP protocol.<br>**Note:** MCTP address of the SMT logic. All MCTP protocol transactions targeting this address will be ACKd by the HW if the TPOLICY.MCTP_EN register is set. |
| 60 | + F0h | 31:16 | IE SMBus Management Transport (SMT) 2 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Notes:**<br>-To be configured in conjunction with the ASF UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 46 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 61 | + F4h | 15:0 | IE SMBus Management Transport (SMT) 2 - Alert Standard Format (ASF) Unique Device Identifier (UDID) B118 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the ASF UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Notes:**<br>-To be configured in conjunction with the ASF UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 61 | + F4h | 31:16 | IE SMBus Management Transport (SMT) 2 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Vendor ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Vendor ID: Enter a 16 bit subsystem Vendor ID<br>**Notes:**<br>-To be configured in conjunction with the MCTP UDID Subsystem Device ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |
| 62 | + F8h | 15:0 | IE SMBus Management Transport (SMT) 2 - MCTP Unique Device Identifier (UDID) B1512 (Subsystem Device ID) | Value: Hex | 0x0000 | Tells the SoC the MCTP UDID Subsystem Device ID: Enter a 16 bit subsystem Device ID<br>**Notes:**<br>-To be configured in conjunction with the MCTP UDID Subsystem Vendor ID Soft Strap<br>-The master uses this to query devices on the SMBus network. |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 47 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| **Reserved** | | | | | | |
| 0 | + 0h | 31:7 | RESERVED | N/A | 0x0000 | Reserved |
| 1 | + 4h | 0 | RESERVED | N/A | 0x1 | Reserved |
| 1 | + 4h | 4:1 | RESERVED | N/A | 0x0 | Reserved |
| 1 | + 4h | 5 | RESERVED | N/A | 0x1 | Reserved |
| 1 | + 4h | 9:6 | RESERVED | N/A | 0x1 | Reserved |
| 1 | + 4h | 10 | RESERVED | N/A | 0x1 | Reserved |
| 1 | + 4h | 14:11 | RESERVED | N/A | 0x3 | Reserved |
| 1 | + 4h | 19:16 | RESERVED | N/A | 0x2 | Reserved |
| 1 | + 4h | 30:27 | RESERVED | N/A | 0x00 | Reserved |
| 1 | + 4h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 2 | + 8h | 31:1 | RESERVED | N/A | 0x00000000 | Reserved |
| 3 | + Ch | 31:1 | RESERVED | N/A | 0x00000000 | Reserved |
| 4 | + 10h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 1 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 2 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 3 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 4 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 5 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 6 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 7 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 11:10 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 12 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 13 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 14 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 48 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 4 | + 10h | 15 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 18:16 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 19 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 20 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 22:21 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 23 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 24 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 25 | RESERVED | N/A | 0x1 | Reserved |
| 4 | + 10h | 26 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 27 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 28 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 29 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 30 | RESERVED | N/A | 0x0 | Reserved |
| 4 | + 10h | 31 | RESERVED | N/A | 0x1 | Reserved |
| 5 | + 14h | 9 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 10 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 11 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 12 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 13 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 14 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 16 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 18 | RESERVED | N/A | 0x0 | Reserved |
| 5 | + 14h | 31:21 | RESERVED | N/A | 0x000 | Reserved |
| 6 | + 18h | 3:0 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 49 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 6 | + 18h | 7:4 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 15:12 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 23:18 | RESERVED | N/A | 0x00 | Reserved |
| 6 | + 18h | 24 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 25 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 26 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 27 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 28 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 30 | RESERVED | N/A | 0x0 | Reserved |
| 6 | + 18h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 7 | + 1Ch | 0 | RESERVED | N/A | 0x0 | Reserved |
| 7 | + 1Ch | 3:2 | RESERVED | N/A | 0x0 | Reserved |
| 7 | + 1Ch | 31:8 | RESERVED | N/A | 0x000000 00 | Reserved |
| 8 | + 20h | 15:0 | RESERVED | N/A | 0x0000 | Reserved |
| 8 | + 20h | 23:16 | RESERVED | N/A | 0x77 | Reserved |
| 8 | + 20h | 31:24 | RESERVED | N/A | 0x78 | Reserved |
| 9 | + 24h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 14 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 16 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 19:17 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 22:20 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 23 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 24 | RESERVED | N/A | 0x0 | Reserved |
| 9 | + 24h | 28 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 50 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 9 | + 24h | 29 | RESERVED | N/A | 0x0 | Reserved |
| 10 | + 28h | 3:0 | RESERVED | N/A | 0x5 | Reserved |
| 10 | + 28h | 6:4 | RESERVED | N/A | 0x0 | Reserved |
| 10 | + 28h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 10 | + 28h | 13:11 | RESERVED | N/A | 0x00 | Reserved |
| 10 | + 28h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 10 | + 28h | 23:16 | RESERVED | N/A | 0x00 | Reserved |
| 10 | + 28h | 31:24 | RESERVED | N/A | 0x00 | Reserved |
| 12 | + 30h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 1 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 2 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 5:3 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 7:6 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 8 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 9 | RESERVED | N/A | 0x1 | Reserved |
| 12 | + 30h | 11:10 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 12 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 15:13 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 18:16 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 20:19 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 21 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 22 | RESERVED | N/A | 0x1 | Reserved |
| 12 | + 30h | 23 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 24 | RESERVED | N/A | 0x0 | Reserved |
| 12 | + 30h | 25 | RESERVED | N/A | 0x1 | Reserved |
| 12 | + 30h | 28:26 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 51 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 12 | + 30h | 31:29 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 1 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 7:2 | RESERVED | N/A | 0x00 | Reserved |
| 13 | + 34h | 8 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 11:9 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 14:12 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 18:16 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 19 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 21:20 | RESERVED | N/A | 0x1 | Reserved |
| 13 | + 34h | 22 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 23 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 26:24 | RESERVED | N/A | 0x0 | Reserved |
| 13 | + 34h | 31:27 | RESERVED | N/A | 0x00 | Reserved |
| 14 | + 38h | 0 | RESERVED | N/A | 0x0000 | Reserved |
| 14 | + 38h | 20:18 | RESERVED | N/A | 0x0 | Reserved |
| 14 | + 38h | 31:22 | RESERVED | N/A | 0x000 | Reserved |
| 15 | + 3Ch | 0 | RESERVED | N/A | 0x1 | Reserved |
| 15 | + 3Ch | 1 | RESERVED | N/A | 0x0 | Reserved |
| 15 | + 3Ch | 7:3 | RESERVED | N/A | 0x00 | Reserved |
| 15 | + 3Ch | 15:13 | RESERVED | N/A | 0x0 | Reserved |
| 16 | + 40h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 17 | + 44h | 31:7 | RESERVED | N/A | 0x0000000 | Reserved |
| 18 | + 48h | 31:0 | RESERVED | N/A | 0x0000000 00 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 52 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 19 | + 4Ch | 1:0 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 2 | RESERVED | N/A | 0x1 | Reserved |
| 19 | + 4Ch | 5:3 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 6 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 7 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 15:14 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 16 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 17 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 18 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 19 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 20 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 21 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 23:22 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 24 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 25 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 26 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 27 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 28 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 29 | RESERVED | N/A | 0x0 | Reserved |
| 19 | + 4Ch | 31:30 | RESERVED | N/A | 0x0 | Reserved |
| 20 | + 50h | 17:16 | RESERVED | N/A | 0x0 | Reserved |
| 20 | + 50h | 19:18 | RESERVED | N/A | 0x0 | Reserved |
| 20 | + 50h | 21:20 | RESERVED | N/A | 0x0 | Reserved |
| 20 | + 50h | 23:22 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 0 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 53 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 21 | + 54h | 1 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 2 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 3 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 4 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 5 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 6 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 8 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 9 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 10 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 11 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 12 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 13 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 14 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 21 | + 54h | 31:26 | RESERVED | N/A | 0x0 | Reserved |
| 22 | + 58h | 6:4 | RESERVED | N/A | 0x0 | Reserved |
| 22 | + 58h | 9:7 | RESERVED | N/A | 0x0 | Reserved |
| 22 | + 58h | 12:10 | RESERVED | N/A | 0x0 | Reserved |
| 22 | + 58h | 15:13 | RESERVED | N/A | 0x0 | Reserved |
| 22 | + 58h | 17:16 | RESERVED | N/A | 0x2 | Reserved |
| 22 | + 58h | 19:18 | RESERVED | N/A | 0x2 | Reserved |
| 22 | + 58h | 21:20 | RESERVED | N/A | 0x2 | Reserved |
| 22 | + 58h | 23:22 | RESERVED | N/A | 0x2 | Reserved |
| 23 | + 5Ch | 9:8 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 54 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 23 | + 5Ch | 10 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 11 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 12 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 13 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 14 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 15 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 16 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 17 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 18 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 19 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 20 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 21 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 22 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 23 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 24 | RESERVED | N/A | 0x0 | Reserved |
| 23 | + 5Ch | 31:29 | RESERVED | N/A | 0x0 | Reserved |
| 24 | + 60h | 6:4 | RESERVED | N/A | 0x0 | Reserved |
| 24 | + 60h | 9:7 | RESERVED | N/A | 0x0 | Reserved |
| 24 | + 60h | 12:10 | RESERVED | N/A | 0x0 | Reserved |
| 24 | + 60h | 15:13 | RESERVED | N/A | 0x0 | Reserved |
| 24 | + 60h | 17:16 | RESERVED | N/A | 0x2 | Reserved |
| 24 | + 60h | 19:18 | RESERVED | N/A | 0x2 | Reserved |
| 24 | + 60h | 21:20 | RESERVED | N/A | 0x2 | Reserved |
| 24 | + 60h | 23:22 | RESERVED | N/A | 0x2 | Reserved |
| 25 | + 64h | 9:8 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 55 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 25 | + 64h | 10 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 11 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 12 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 13 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 14 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 16 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 17 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 18 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 19 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 20 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 21 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 22 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 23 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 24 | RESERVED | N/A | 0x0 | Reserved |
| 25 | + 64h | 31:29 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 16 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 17 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 18 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 19 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 20 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 21 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 22 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 23 | RESERVED | N/A | 0x0 | Reserved |
| 26 | + 68h | 31:24 | RESERVED | N/A | 0x00 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 56 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 27 | + 6Ch | 16 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 17 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 18 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 19 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 20 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 21 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 22 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 23 | RESERVED | N/A | 0x0 | Reserved |
| 27 | + 6Ch | 31:24 | RESERVED | N/A | 0x00 | Reserved |
| 28 | + 70h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 28 | + 70h | 1 | RESERVED | N/A | 0x0 | Reserved |
| 28 | + 70h | 2 | RESERVED | N/A | 0x0 | Reserved |
| 28 | + 70h | 3 | RESERVED | N/A | 0x0 | Reserved |
| 28 | + 70h | 31:4 | RESERVED | N/A | 0x0000000 | Reserved |
| 29 | + 74h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 29 | + 74h | 1 | RESERVED | N/A | 0x0 | Reserved |
| 29 | + 74h | 2 | RESERVED | N/A | 0x0 | Reserved |
| 29 | + 74h | 3 | RESERVED | N/A | 0x0 | Reserved |
| 29 | + 74h | 31:4 | RESERVED | N/A | 0x0000000 | Reserved |
| 30 | + 78h | 31:2 | RESERVED | N/A | 0x0000000 | Reserved |
| 31 | + 7Ch | 0 | RESERVED | N/A | 0x0 | Reserved |
| 31 | + 7Ch | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 31 | + 7Ch | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 31 | + 7Ch | 31 | RESERVED | N/A | 0x0 | Reserved |
| 32 | + 80h | 7 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 57 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 32 | + 80h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 32 | + 80h | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 32 | + 80h | 31:25 | RESERVED | N/A | 0x00 | Reserved |
| 33 | + 84h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 33 | + 84h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 35 | + 8Ch | 31:18 | RESERVED | N/A | 0x0000 | Reserved |
| 36 | + 90h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 36 | + 90h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 36 | + 90h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 36 | + 90h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 37 | + 94h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 37 | + 94h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 37 | + 94h | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 37 | + 94h | 31:25 | RESERVED | N/A | 0x00 | Reserved |
| 38 | + 98h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 38 | + 98h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 40 | + A0h | 31:18 | RESERVED | N/A | 0x0000 | Reserved |
| 41 | + A4h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 41 | + A4h | 7:1 | RESERVED | N/A | 0x0000 | Reserved |
| 41 | + A4h | 15:9 | RESERVED | N/A | 0x0000 | Reserved |
| 41 | + A4h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 42 | + A8h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 42 | + A8h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 42 | + A8h | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 42 | + A8h | 31:25 | RESERVED | N/A | 0x00 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 58 of 59)

| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 43 | + ACh | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 43 | + ACh | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 45 | + B4h | 31:18 | RESERVED | N/A | 0x0000 | Reserved |
| 46 | + B8h | 31:0 | RESERVED | N/A | 0x00000000 | Reserved |
| 47 | + BCh | 0 | RESERVED | N/A | 0x0 | Reserved |
| 47 | + BCh | 1 | RESERVED | N/A | 0x0 | Reserved |
| 47 | + BCh | 12:2 | RESERVED | N/A | 0x000 | Reserved |
| 47 | + BCh | 31:15 | RESERVED | N/A | 0x00000 | Reserved |
| 48 | + C0h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 48 | + C0h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 48 | + C0h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 48 | + C0h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 49 | + C4h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 49 | + C4h | 15 | RESERVED | N/A | 0x0 | Reserved |
| 49 | + C4h | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 49 | + C4h | 31:25 | RESERVED | N/A | 0x00 | Reserved |
| 50 | + C8h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 50 | + C8h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 52 | + D0h | 31:18 | RESERVED | N/A | 0x0000 | Reserved |
| 53 | + D4h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 53 | + D4h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 53 | + D4h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 53 | + D4h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 54 | + D8h | 7 | RESERVED | N/A | 0x0 | Reserved |
| 54 | + D8h | 15 | RESERVED | N/A | 0x0 | Reserved |

## Table 4-3. Flash Descriptor Soft Straps (Sheet 59 of 59)

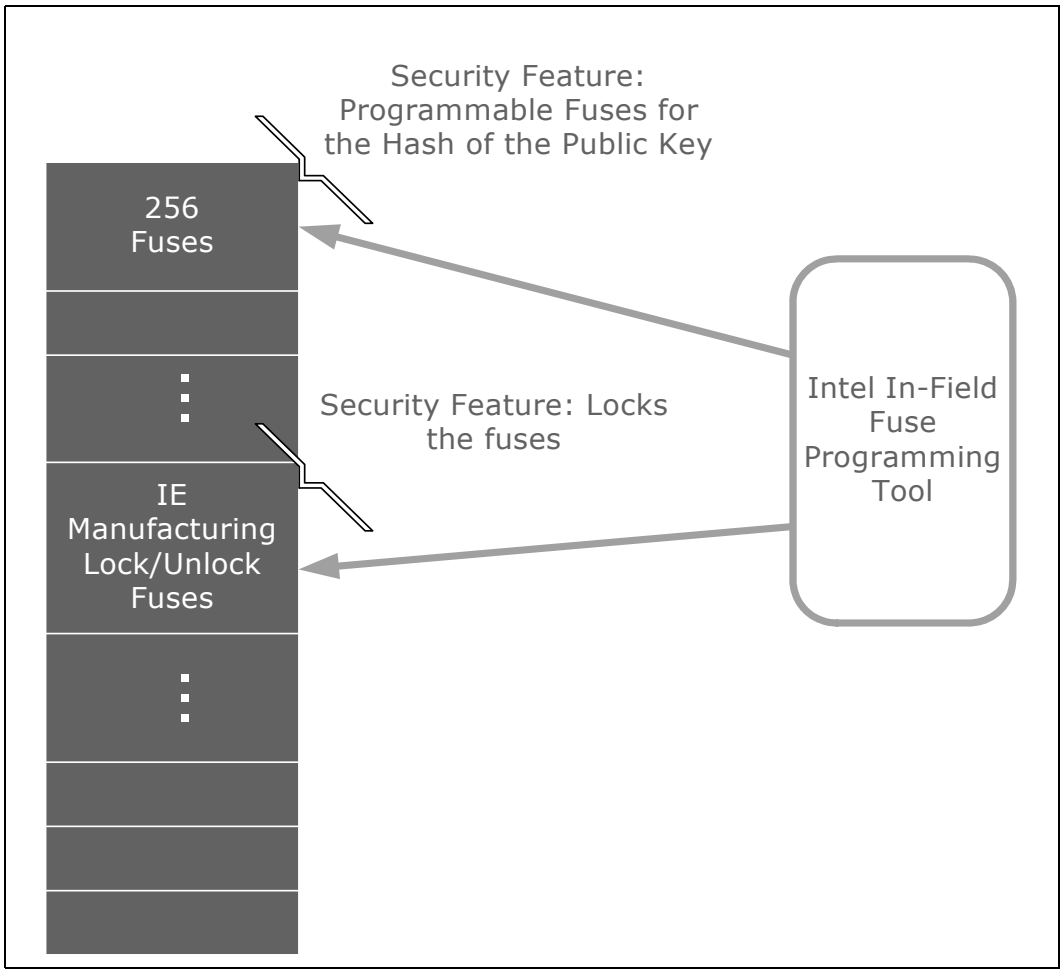
| FITC SoC SS # | FPSBA + Offset | Bits | Soft Strap Name | Soft Strap Value List | Default Value | Description |
|---|---|---|---|---|---|---|
| 54 | + D8h | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 54 | + D8h | 31:25 | RESERVED | N/A | 0x00 | Reserved |
| 55 | + DCh | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 55 | + DCh | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 57 | + E4h | 31:18 | RESERVED | N/A | 0x0000 | Reserved |
| 58 | + E8h | 0 | RESERVED | N/A | 0x0 | Reserved |
| 58 | + E8h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 58 | + E8h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 58 | + E8h | 31 | RESERVED | N/A | 0x0 | Reserved |
| 59 | + ECh | 7 | RESERVED | N/A | 0x0 | Reserved |
| 59 | + ECh | 15 | RESERVED | N/A | 0x00 | Reserved |
| 59 | + ECh | 23:17 | RESERVED | N/A | 0x00 | Reserved |
| 59 | + ECh | 31:25 | RESERVED | N/A | 0x00 | Reserved |
| 60 | + F0h | 7:1 | RESERVED | N/A | 0x00 | Reserved |
| 60 | + F0h | 15:9 | RESERVED | N/A | 0x00 | Reserved |
| 62 | + F8h | 31:18 | RESERVED | N/A | 0x0000 | Reserved |

## 4.3 Fuses Pertaining to the Innovation Engine (IE)

***Note:*** **Intel recommends that customers perform the IE manufacturing lock step described below, even if not using the Innovation Engine, to eliminate any in field security risk.**

There is a set of fuses pertaining to the Innovation Engine (IE) that must be configured with an Intel in-field fuse programming tool. See Figure 4-3, "IE Fuses." These fuses have to be programmed as a security measure to set the hash of the public cryptographic key by the OEM in order for them to use the IE. After the hash of the public key is set, the tool will complete an IE manufacturing lock process flow to finalize the IE. For more information, refer to Chapter 37, "Innovation Engine."

**Figure 4-3. IE Fuses**




§ §

# 5 System Agent and Root Complex

## 5.1 Introduction

The SoC contains a System Agent block that provides the main interface with the processor cores and the other integrated elements. The I/O Fabric provides the connections to the PCI Express* Root Ports and the integrated I/O devices.

The system agent and memory fabric includes the coherence engine for handling all the coherent transactions, data-path to merge coherent and non-coherent traffic, write combining buffers to merge both coherent and non-coherent partial writes and provide the single point of global observation for all transactions. The coherent and memory fabric interfaces to I/O devices through a standard I/O Fabric root complex.

**Figure 5-1. What is Covered in This Chapter**

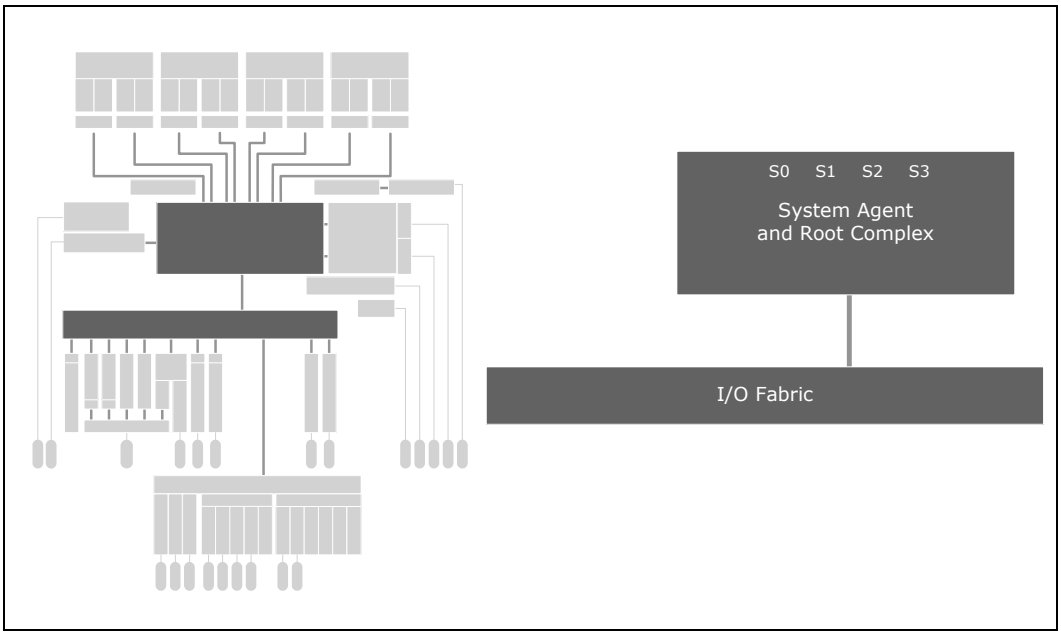



**Table 5-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| PCI Express* | 3.1 | March 4, 2009 | *PCI Express Base Specification Revision 3.0* |
| Software Developer's Manual (SDM) | - | - | *Intel® 64 and IA-32 Architectures Software Developer's Manual* |

## 5.2 Features

- 40-bit physical, 48-bit linear memory-space addressing
- 20 programmable isolated memory regions (IMR)
- Patrol scrub engine performs a DDR memory scrub to fix correctable memory errors in the background
- Support for xAPIC interrupt architecture:
  - Legacy interrupts (SMI#, INIT#, INTR#, NMI) from IO agents
- Support for x2APIC:
  - In x2APIC mode, APIC registers are accessed through Model Specific Register (MSR) interfaces.
- PCI Express Advanced Error Reporting (AER) support
- MSI signaling
- INTx signaling
- Internal command and data path parity coverage
- Internal RAM parity coverage

## 5.3 Root Complex

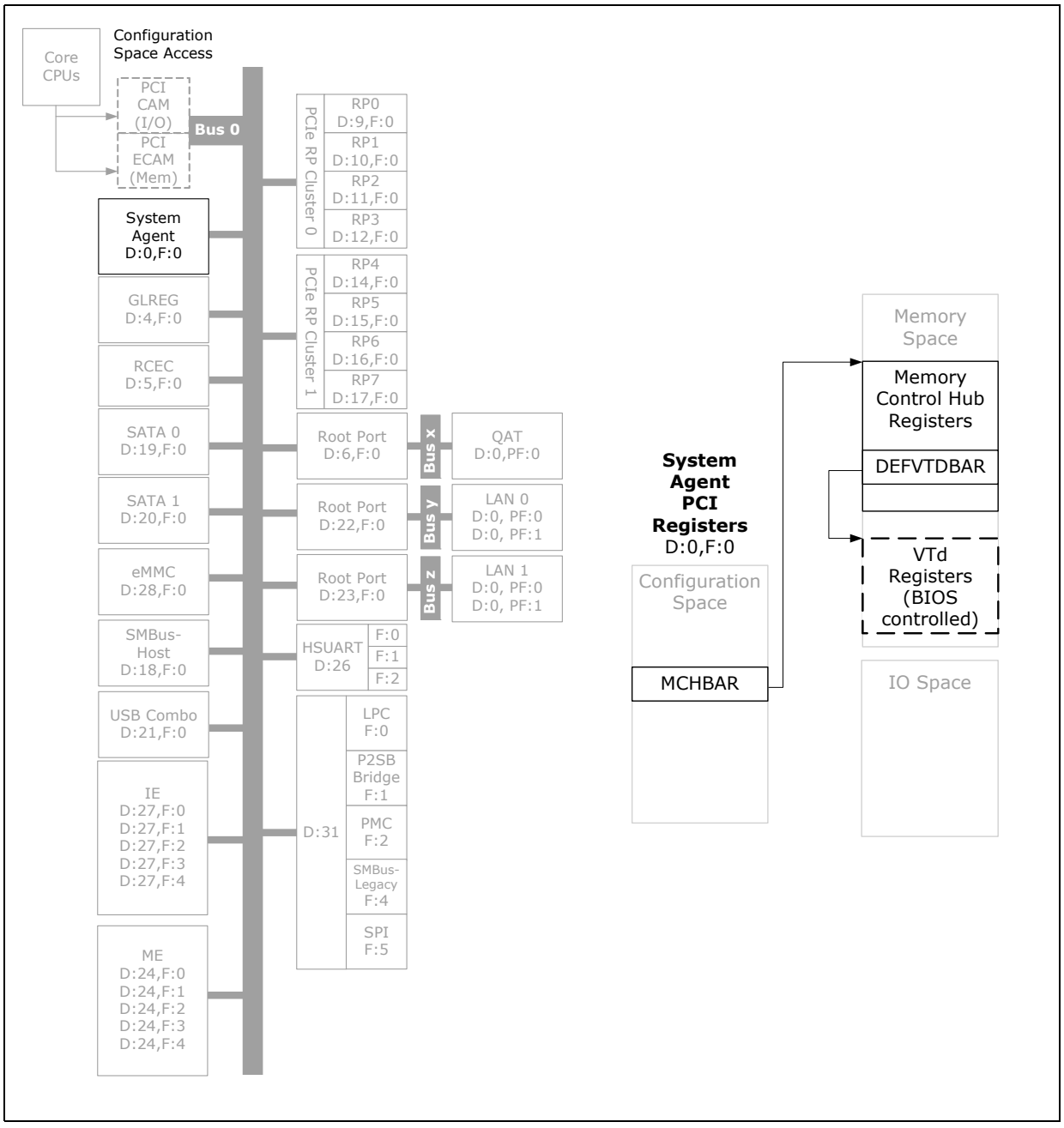
The Host Root Complex, also called the Primary Scalable Fabric (PSF), implements the bus 0 interconnect of a PCI Express* Root Complex. The root complex contains the internal Root Complex Event Collector (RCEC) which is a sub-agent of the Primary to Side Band Bridge (P2SB). The Root Complex is also where the Global and Local Error control and reporting are located as well as most of the SoC Reliability, Availability, and Serviceability (RAS) capabilities. See Chapter 6, “Reliability, Availability and Serviceability (RAS)”.

There is a PCI hierarchy for the Host/IA cores, a hierarchy for Innovation Engine (IE), and one for the Intel® Management Engine (Intel® ME).

## 5.4 Register Map

Figure 5-2 shows the associated registers from a system software viewpoint.

**Figure 5-2. Register Map**



### 5.4.1 Registers in Configuration Space

**Table 5-2. Registers in Host Configuration Space**

| Configuration Space of B0:D0:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | DEVICE_ID_VENDOR_ID_0_0_0_PCI | Device ID and Vendor ID Register |
| 4 | PCI_STATUS_COMMAND_0_0_0_PCI | PCI Status and PCI Command Register |
| 8 | REVISION_ID_CLASS_CODE_0_0_0_PCI | PCI Revision ID and PCI Class Code Register |
| C | MASTER_LATENCY_TIME_0_0_0_PCI | Master Latency Timer and Header Type Register |
| 2C | SVID_SID_0_0_0_PCI | PCI Subsystem Vendor ID and PCI Subsystem ID |
| 34 | CAPPTR_0_0_0_PCI | Capability Register Pointer |
| 48 | MCHBAR_LO_0_0_0_PCI | Memory Controller Hub Base Address Register |
| 4C | MCHBAR_HI_0_0_0_PCI | Memory Controller Hub Base Address Register |
| 60 | PCIEXBAR_LO_0_0_0_PCI | PCI Express Enhanced Configuration Range Base Address Low |
| 64 | PCIEXBAR_HI_0_0_0_PCI | PCI Express Enhanced Configuration Range Base Address High |
| A8 | TOUUD_LO_0_0_0_PCI | Top of Upper Usable DRAM Low |
| AC | TOUUD_HI_0_0_0_PCI | Top of Upper Usable DRAM High |
| BC | TOLUD_0_0_0_PCI | Top of Lower Usable DRAM |
| E0 | CAPID0_CAPCTRL0_0_0_0_PCI | Capability ID0 Capability Control |
| E4 | CAPID0_A_0_0_0_PCI | Capability ID0 A |

§ §

# 6 Reliability, Availability and Serviceability (RAS)

## 6.1 SoC RAS Overview

This chapter describes the features provided by the SoC for the development of high RAS (Reliability, Availability, Serviceability) systems. RAS refers to three main features associated with system's robustness. These features are summarized as follows:

- **Reliability:** Refers to how often errors occur in the system, and whether the system can recover from an error condition.
- **Availability:** Refers to how flexible the system resources can be allocated or redistributed for the system utilizations and system recovery from errors.
- **Serviceability:** Refers to how well the system reports and handles events related to errors.

The SoC is designed to approach the RAS capabilities of Intel® Xeon® processors designed for servers.

## 6.2 Signal Descriptions

The signal descriptions pertaining to this chapter are shown in Table 6-1, "Signal Names and Descriptions". For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 26, "Customer General-Purpose I/O (GPIO)". The Direction/Type column of Table 6-1, "Signal Names and Descriptions" is interpreted as follows:

- O = Output signal. The SoC has driver circuit.

**Table 6-1. Signal Names and Descriptions**

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| ERROR_N[0]<br>ERROR_N[1]<br>ERROR_N[2] | O | Yes | **Error:** These active-low signals indicate to the external circuitry the severity of a detected error:<br>• ERROR_N[0] - Correctable Error<br>• ERROR_N[1] - Non-Fatal Error<br>• ERROR_N[2] - Fatal Error<br>While the SoC active-low output signal PMU_PLTRST_N is asserted, these signals are not valid and must be ignored by the platform board circuitry. |
| IERR_N | O | Yes | **Internal Error:** This active-low signal indicates to the external circuitry that the SoC has detected an error.<br>While the SoC active-low output signal PMU_PLTRST_N is asserted, this signal is not valid and must be ignored by the platform board circuitry. |
| MCERR_N | O | Yes | **Machine-Check Error:** This active-low signal indicates to the external circuitry that the SoC has detected a Machine Check error.<br>While the SoC active-low output signal PMU_PLTRST_N is asserted, this signal is not valid and must be ignored by the platform board circuitry. |

**Table 6-2. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| Software Developer's Manual | - | - | *Intel® 64 and IA-32 Architectures Software Developer's Manual* |
| PCI Express* Specification | 3.0 | Nov. 10, 2010 | *PCI Express Base Specification Revision 3.0* |
| ACPI Specification | 5.0a | Nov. 13, 2013 | *Advanced Configuration and Power Interface Specification Version 5.0, Errata A* |

## 6.3 Feature Summary

- Machine Check Architecture (MCA) implemented which provides MCA Bank registers for Core, L2 Cache, System Agent, Memory Controller, and Power-Management Unit.
  - Error Cloaking and Dual Signaling (MSR 52h) to optionally hide errors from the operating system
  - Corrected Machine-Check error Interrupt (CMCI) that eliminates polling by generating a local interrupt based on programmable threshold values for accumulated MCA corrected errors
- Parity-protected internal data paths of the System Agent circuitry and the interface between the System Agent and the processor cores/L2-cache.
- Parity-protected command and data for the internal I/O system fabric. Parity errors are logged
- Memory Controllers supports ECC, data scrambling, patrol scrubbing and demand scrubbing
- Advanced Error Reporting (AER) supported which provides OS-level error recovery form system errors and also system debug:
  - PCI Express* Root Ports
  - Root Complex Integrated Endpoints: SMBus-Host function; Global/Local Integrated Error Handler (IEH) function
  - Integrated LAN Controllers
  - Integrated Intel® QuickAssist Technology (Intel® QAT) Controller
  - PCI Express* Root Ports (but not the Virtual Root Ports)
- Virtual partitioning with:
  - Enhanced Intel® Virtualization Technology (Intel® VT) for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x)
  - Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)
- Error Injection for validation of system error-handling software

## 6.4 Architectural Overview

This RAS chapter provides information related to the SoC error detection and error reporting mechanisms. The SoC provides Machine Check Architecture to detect and report hardware errors in the CPU cores, L2 Cache, system memory, and the SoC System Agent. The SoC also provides a set of Global Error Registers that are based on a number of Local Error Registers to detect and report SoC-internal hardware errors as well as hardware errors detected by the integrated I/O devices. A number of these I/O devices also support Advanced Error Reporting (AER) which is an optional PCI Express* Extended Capability.

In some cases, these mechanisms are configured to generate interrupts or system events.

Some SoC internal controllers provide ways of maintaining high levels of system availability when hardware errors are detected. Error-correcting logic, transaction retries, and degraded operation are examples.

### 6.4.1 Error Classification

The ACPI and PCIe* specifications classify errors. This classification separates hardware errors that result in degraded performance from those that result in functional failures:

- Correctable Errors
- Uncorrectable Errors

Uncorrectable hardware errors can be further classified as:

  — Non-Fatal Error

  — Fatal Error

In many cases, just the terms Correctable, Non-Fatal and Fatal are used to classify hardware errors.

### 6.4.2 Error Detection and Reporting Mechanisms

Machine Check Errors are defined for the SoC. See Section 6.6, "Machine Check Architecture (MCA)". They are accessible to software via Model Specific Registers (MSRs). Errors in the CPU cores, L2 cache, and System Agent are reported as Machine Check Error MSRs.

Besides SoC errors reported through the Machine Check Architecture (MCA), local device errors are detected, logged and reported by the SoC through Global Error registers. These registers are accessible to software via Host Root Configuration Space and Host Root Memory Space. System Agent and peripheral I/O device errors are reported in this manner. There are five types of local devices:

- PCIe Root Ports.
- PCIe Endpoints behind the Virtual Root Ports
- PCIe Root Complex Integrated Endpoints
- Legacy PCI devices
- Non-PCI devices

The manner in which each of the five local device types deals with local errors are explained in Section 6.13, "Overview of SoC Error Registers". There is a set of Global Error registers that summarize the Local Error registers and can be configured to generate an SMI or NMI.

In addition to providing software access to error registers, the SoC provides output signal pins to notify external hardware circuitry that an error has been detected. The severity of the error is also provided on signal pins. Descriptions for all five of these output signal pins are in Table 6-1, "Signal Names and Descriptions". External circuitry can use the SoC SMBus-PECI interface to read various internal registers containing error information.

## 6.5 Availability Features

### 6.5.1 System Memory

The SoC Memory Controllers support Single-Bit Error Correction using ECC. The controllers also provide Data Scrambling, Patrol Scrubbing and Demand Scrubbing.

#### 6.5.1.1 Single Bit Error Correction

The memory controllers support ECC. Single-bit errors are corrected and Correctable Errors reported and logged.

#### 6.5.1.2 DRAM Data Scrubbing

Data scrubbing features require the use of Error Correction Codes (ECC) in the system memory design.

Demand Scrubbing writes correct data back to memory for any read request that returns with a correctable data error. The DEMAND_SCRUB_ENABLE field (bit 3) of the B-Unit Control (B_CR_BCTRL2) register enables Demand Scrubbing when set to 1. The B_CR_BCTRL2 register is located in Host Root Memory Space at MCHBAR plus offset 6D70h.

Patrol Scrubbing ensures that data with a Correctable Error does not remain in DRAM long enough to stand a significant chance of further corruption and become an Uncorrectable Error. If enabled, the Patrol Scrubbing generates memory requests at a programmed interval. If a correctable error is detected by a scrubbing request, Demand Scrubbing is used to correct the error. The PAT_SCRUB_ENABLE field (bit 31) of the Patrol Scrub Configuration (DPATROL_SCRUB_CFG) register enables Patrol Scrubbing when set to 1. The DPATROL_SCRUB_CFG register is located in Host Root Memory Space at SBREG_BAR plus offset 10_00B0h. For Memory Controller 1, the DCS register is located at SBREG_BAR, offset 12_00B0h. Various registers first need to be setup by BIOS in order to deploy Patrol Scrubbing.

See Section 6.16, “Error Reporting Flow — Non-PCI devices” for SoC error reporting during scrubbing.

#### 6.5.1.3 DRAM Data Scrambling

As an option, the SoC Memory Controllers provide DRAM Data Scrambling which is a technique to reduce noise on the internal power-supply rails and improve signal integrity. Across the 72 Data/ECC bits of the DQ/ECC bus, it drives very close to 50% ones and 50% zeros in every cycle. This eliminates the worst-case situation where all bits of the bus would simultaneously drive high or drive low.

Data Scrambling is available for both ECC and non-ECC memory platform designs.

Form Memory Controller 0, the SCRAMBLE_ENABLE field (bit 23) of the DRAM Control and Status (DCS) register, when set to 1, the data lanes (DDR0_DQ[63:0]) and the ECC lanes (DDR0_ECC[7:0]) will be scrambled, otherwise no scrambling will occur. The DCS register is located in Host Root Memory Space at SBREG_BAR plus offset 10_0170h. For Memory Controller 1, the DCS register is located at SBREG_BAR, offset 12_0170h.

### 6.5.2 PCI Express Root Ports

The *PCI Express Base Specification*, Revision 3.0 defines a standard set of error-reporting mechanisms. The SoC supports them all including Error Poisoning and Advanced Error Reporting (AER). The error reporting mechanism and the error registers are described in Section 6.12, "SoC UnCore and I/O Error Reporting".

Besides error detection, logging and reporting, the SoC PCI Express Root Ports support:

- Data Link Layer 32-bit CRC (LCRC) and Retry Management — see sub-section 3.5 "Data Integrity" of the Data Link Layer section in the *PCI Express Base Specification*, Revision 3.0.
- Transaction Layer end-to-end 32-bit CRC (ECRC) — see sub-section 2.7 "Data Integrity" of the Transaction Layer section in the *PCI Express Base Specification*, Revision 3.0.
  - — Supported by the PCIe Root Ports but not on the three Virtual Root Ports.
- Link Retraining and Dynamic Link Width Reduction upon link failure.
  - — Supported by the PCIe Root Ports but does not apply to the three Virtual Root Ports.

When PCIe Link Retraining fails, the SoC PCIe interface provides a mechanism to recover from the failed link. The PCIe link is capable of operating in different link widths. The SoC supports PCIe Root Port operation in x8, x4, and x2. In case of a persistent link failure, the PCIe link can fall back to a smaller link width, as small as x1, in attempt to recover from the error. As an example, a PCIe x8 link can fall back to a x4 link. If necessary, it can be further reduced to a x2 link, and then to X1 link. This mechanism enables continuation of system operation in case of PCIe link failures. Additional information can be found in Section 11.4.5, "Degraded Mode".

## 6.6 Machine Check Architecture (MCA)

Machine Check Architecture is a mechanism for detecting and reporting hardware (machine) errors in the CPU and UnCore. For the SoC, MCA is not used to report I/O-based errors.

This section provides the necessary details for the operating software to handle Machine Check Exceptions (MCE). Some operating systems hook the Machine Check Architecture (MCA) exception vector (18h) to allow system-crash analysis. Like some other Intel processors, the SoC has been enhanced to allow the machine state to be preserved across the assertion of the Platform Reset. The BIOS does not modify the MCA registers following the assertion of Platform Reset. This allows the operating system to enhance the exception handler by having this information available following a reboot after an error has occurred. Only upon the assertion (the signal transition from low to high) of the COREPWROK input signal (indicating POWERGOOD, power-on) is the machine-check architecture state re-initialized.

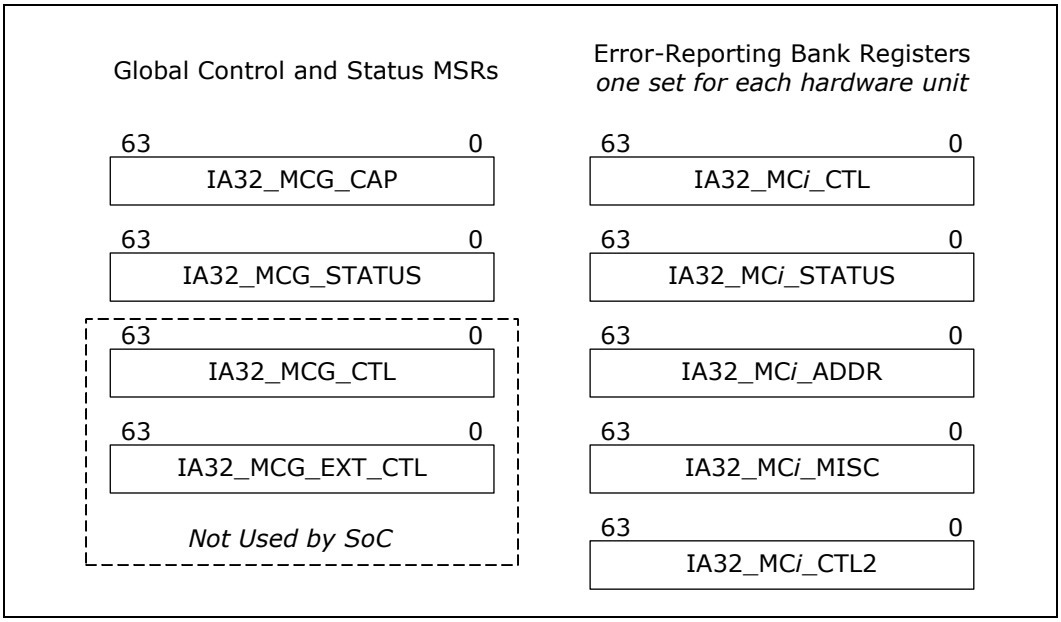
All MCA state information is accessible via the Model-Specific Register (MSR) accesses using the Read MSR (RDMSR) and Write MSR (WRMSR) instructions. RDMSR and WRMSR are described in Volume 2, Chapter 4 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*. The machine-check architecture is described in Volume 3, Chapter 15, of the same manual.

Intel® architecture defines two classifications of MCA MSRs:

1. Machine Check Global Control and Status registers.
2. Machine Check Error-Reporting Bank registers.

The SoC does not implement the IA32_MCG_CTL and IA32_MCG_EXT_CTL Global Control and Status MSRs. See Figure 6-1, "SoC Machine Check MSRs — Global Control and Bank Registers".

**Figure 6-1. SoC Machine Check MSRs — Global Control and Bank Registers**



### 6.6.1 Machine Check MSR Addresses

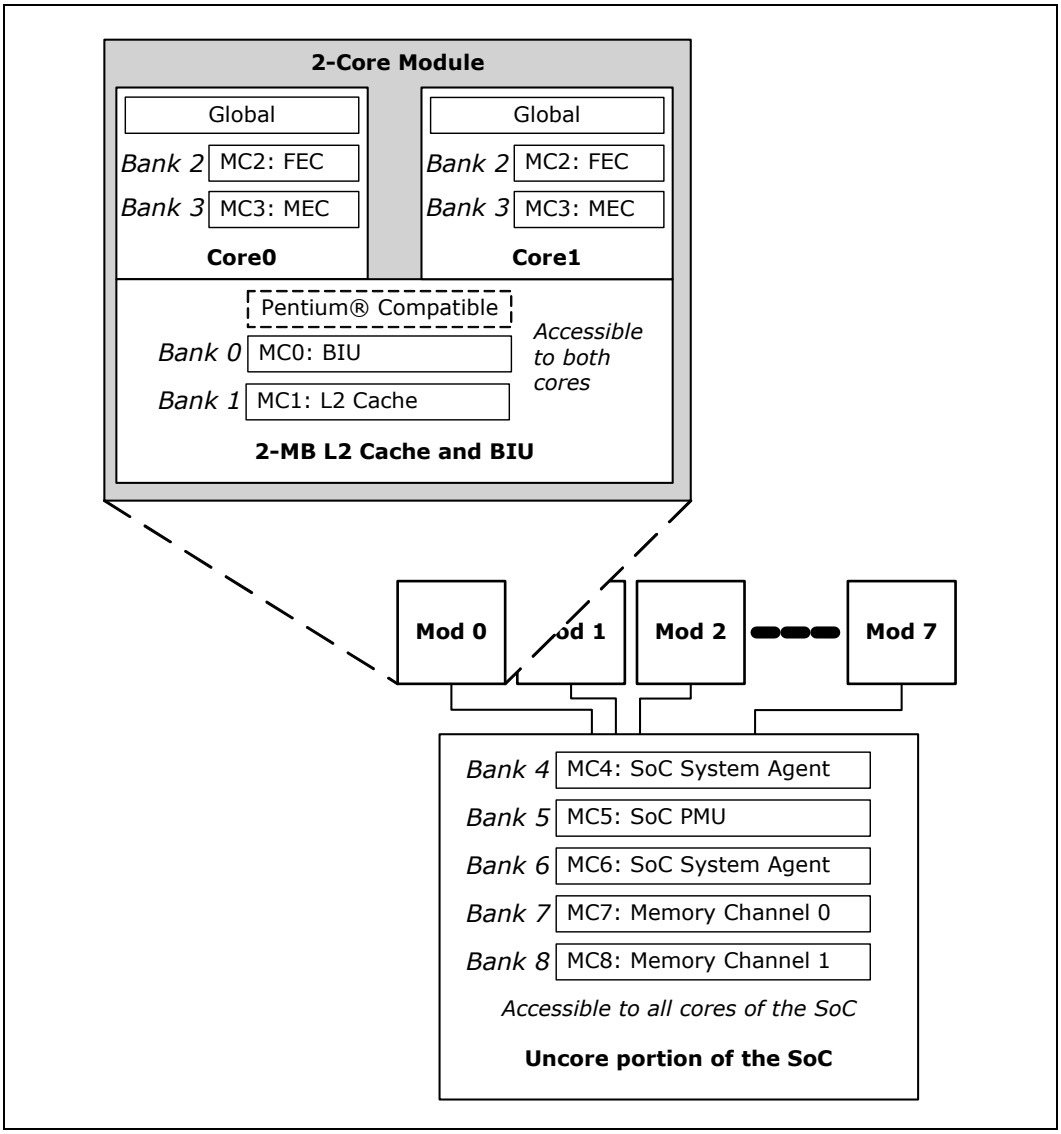
The machine check MSRs are accessed via the RDMSR and WRMSR processor instructions.

Table 6-3 lists the MSR addresses and Figure 6-2, "SoC Core, Module, and Package Machine Check Registers" shows the core, module, and SoC package with respect to each Machine Check register.

**Table 6-3. SoC Machine Check MSR Addresses**

| Register Type | Function | Register | Register Address |
|---|---|---|---|
| MC Bank 0 | Module - Bus Interface Unit (BIU) | IA32_MC0_STATUS<br>IA32_MC0_ADDR | 401h<br>402h |
| MC Bank 1 | Module - L2 Cache | IA32_MC1_STATUS<br>IA32_MC1_ADDR | 405h<br>406h |
| MC Bank 2 | Core - Fetch Execution Cluster (FEC) | IA32_MC2_STATUS<br>IA32_MC2_ADDR | 409h<br>40Ah |
| MC Bank 3 | Core - Memory Execution Cluster (MEC) | IA32_MC3_CTL<br>IA32_MC3_STATUS<br>IA32_MC3_ADDR | 40Ch<br>40Dh<br>40Eh |
| MC Bank 4 | SoC System Agent | IA32_MC4_STATUS<br>IA32_MC4_ADDR<br>IA32_MC4_MISC | 411h<br>412h<br>413h |
| MC Bank 5 | SoC Power Management Unit (PMU) | IA32_MC5_STATUS<br>IA32_MC5_ADDR<br>IA32_MC5_MISC | 415h<br>416h<br>417h |
| MC Bank 6 | SoC System Agent | IA32_MC6_STATUS<br>IA32_MC6_ADDR | 419h<br>41Ah |
| MC Bank 7 | Memory Channel 0 | IA32_MC7_STATUS<br>IA32_MC7_MISC | 41Dh<br>41Fh |
| MC Bank 8 | Memory Channel 1 | IA32_MC8_STATUS<br>IA32_MC8_MISC | 421h<br>423h |

**Figure 6-2. SoC Core, Module, and Package Machine Check Registers**



## 6.6.2 Enhancements to MCA Error Reporting

The SoC provides support of Error Cloaking and Dual Signaling (MSR 52h) to optionally hide errors from the operating system. Corrected Machine-Check error Interrupt (CMCI) enhancement is also supported.

### 6.6.2.1 Error Cloaking

The Error Cloaking feature allows BIOS to hide Correctable Errors from the Operating System (OS). The Model-Specific Register (MSR) MSR_SMM_MCA_CONTROL (MSR 52h) contains a control bit to mask corrected errors during non-System-Management-Mode (non-SMM) accesses.

Non-SMM reads to MCi_STATUS registers will return 0 instead of the actual status value if, MCi_STATUS RDMSR reports a VALID CORRECTED error (VAL=1,UC=0) and CERR_RD_STATUS_IN_SMM_ONLY (in MSR0x52) is set. See Table 6-4, "MSR_SMM_MCA_CONTROL (MSR 52h) — Enhanced MCA Control".

### 6.6.2.2 Dual Signaling

The Dual Signaling feature allows SMM firmware to handle, and possibly clear-out, errors before the Operating System sees them.

In the MCA flow, if PEND_SMI_ON_MCA bit is set in MSR 52h, the SoC creates a pending System Management Interrupt (SMI) on the core. This flow relies on the MCA broadcast (and consistent opt-in and subsequent pending of SMI) to get the SMI pending on all threads. See Table 6-4, "MSR_SMM_MCA_CONTROL (MSR 52h) — Enhanced MCA Control".

**Table 6-4. MSR_SMM_MCA_CONTROL (MSR 52h) — Enhanced MCA Control**

| Bit | Register Field | Access Type | Description |
|---|---|---|---|
| 31:10 | Reserved | - | Reserved |
| 9 | PEND_SMI_ON_MCA | RW | Pend SMI on MCA |
| 8:1 | | - | Reserved |
| 0 | CERR_RD_STATUS_IN_SMM_ONLY | RW | When set to 1, Valid (VAL=1), Correctable Error (UC=0) CPU MCA Status can only be read when the CPU is in System Management Mode (SMM).<br>Otherwise read 0 if VAL=1 and UC=0.<br>This bit is package-scope and mapped by microcode to an UnCore SCP. |

### 6.6.2.3 Corrected Machine-Check error Interrupt (CMCI)

CMCI is an architectural enhancement to the MCA. It provides capabilities beyond those of threshold-based error reporting. With threshold-based error reporting, software is limited to use periodic polling to query the status of hardware corrected Machine-Check errors.

CMCI provides a signaling mechanism to deliver a local interrupt based on threshold values that software can program in bits [14:0] of the Machine-Check Bank IA32_MCi_CTL2 MSRs.

CMCI is disabled by default. The MCG_CMCI_P bit of the IA32_MCG_CAP Global Machine Check register, when set, indicates that the SoC has the CMCI capability. Bit 30 of each IA32_MCi_CTL2 register must be checked to see whether the particular Machine-Check Bank has the CMCI-signaling capability. CMCI is not affected by the CR4.MCE bit, and it is not affected by the IA32_MCi_CTL MSRs. The L2 Cache, SoC internal fabric, and Memory Controller Machine-Check Banks support CMCI signaling.

Corrected Machine-Check Errors detected in the UnCore portion of the SoC (see Figure 6-2, "SoC Core, Module, and Package Machine Check Registers") generate a CMCI to all processor cores of the SoC. There is no provision for one processor to signal a CMCI to other processor within a system.

The processor's APIC Local Vector Table (LVT) has a LVT CMCI register that specifies interrupt Delivery Mode (NMI, SMI, etc.) and Vector when an overflow condition of corrected machine check error count reaching a threshold value occurred in a machine check bank supporting CMCI.

# 6.7 Advanced Error Reporting (AER)

## 6.7.1 AER and the PCIe Root Ports and RCEC

Advanced Error Reporting (AER) is supported on these integrated devices:

- PCI Express* Root Ports.
- Root Complex Integrated Endpoints:
  - — SMBus-Host function
  - — Host Bridge (Global/Local Error function)
- Integrated LAN Controllers
- Integrated Intel® QuickAssist Technology (Intel® QAT) Controller

The RCEC and the PCIe Root Ports with the AER capability not only generate system errors (Correctable, Non-Fatal, and Fatal) from received, or internally-generated, error Messages, each can generate an interrupt for each of the three error categories.

The RCEC_ROOTERRCMD register (ROOTERRCMD for each of the eight Root Ports and three Virtual Root Ports) in Host Root Configuration Space provides an enable/disable for each of the three error categories.

For both Root Ports and Root Complex Event Collectors, in order for a received error Message or an internally generated error Message to generate an interrupt enabled by this register, the error Message must be enabled for "transmission" by the Root Port or Root Complex Event Collector.

The interrupt generated to the host is either an INTx message or an MSI/MSI-X interrupt, depending which type of interrupt-reporting mechanism is enabled for the RCEC or Root Port.

The RCEC_ROOTERRSTS register (ROOTERRSTS for the Root Ports) reports status of error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL) received by the RCEC (or Root Port), and of errors detected by the RCEC/Root Port itself (which are treated conceptually as if the RCEC/Root Port had sent an error Message to itself). The RCEC provide this only for the two Root Complex Integrated Endpoints is serves — the SMBus-Host function and the Global/Local Integrated Error Handler function.

When an error is received by the RCEC or Root Port, the respective first error bit is Set and the Requester ID is logged in the RCEC_ERRSRCID (ERRSRCID for Root Ports) register.

For additional details, see Section 7.10 "Advanced Error Reporting Capability" of the *PCI Express Base Specification Revision 3.0*.

## 6.8 Intel® Virtualization Technology (Intel® VT)

The SoC provides hardware support for isolating & restricting device accesses to the owner of the partition managing the device. Virtual partitioning is provided as:

- Enhanced Intel® Virtualization Technology (Intel® VT) for IA-32, Intel® 64 and Intel® Architecture (Intel® VT-x)
- Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)

See Section 2.2.2, "Intel® Virtualization Technology".

## 6.9 Error Injection

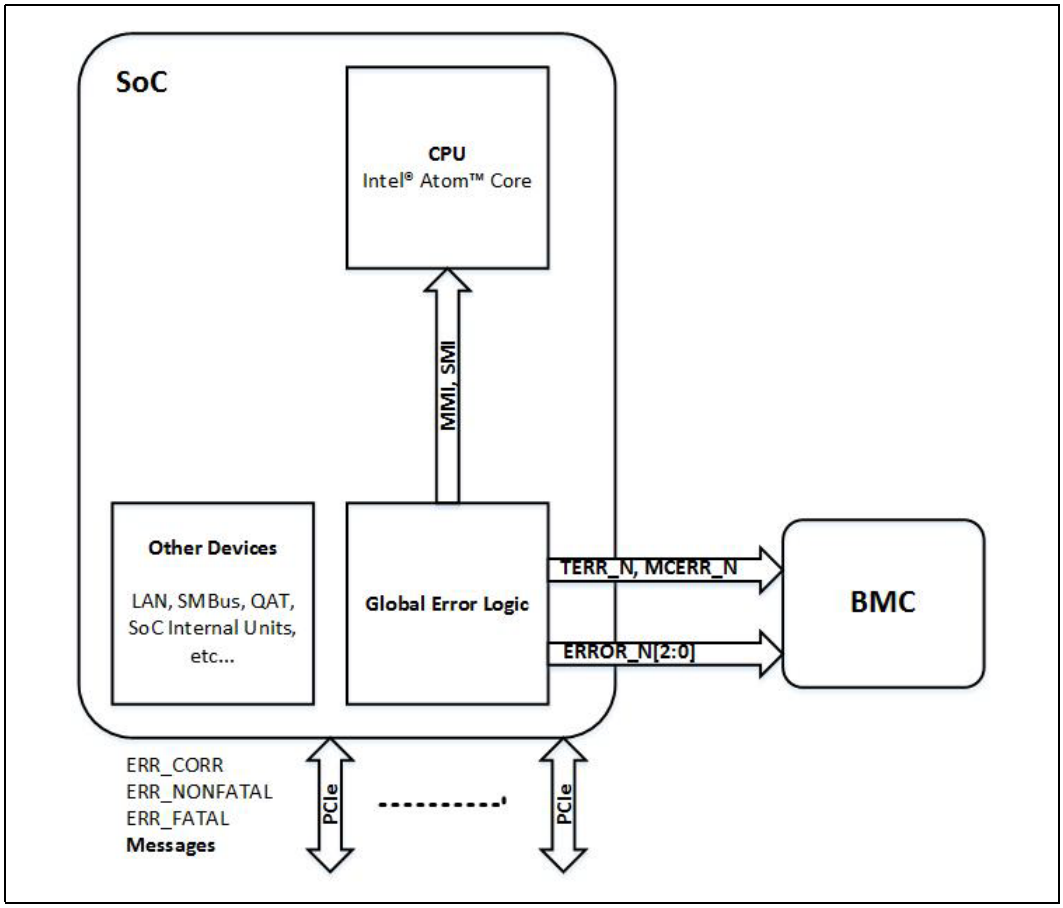
Error injection can be used to validate system error-handling software. The SoC supports:

- Writable Machine Check Architecture (MCA) banks.
- Out-of band access to the MCA registers
- The PCI Express Root Ports support the Windows Hardware Error Architecture (WHEA) capability. The dedicated Virtual Root Ports for the Intel® QuickAssist Technology and LAN Controllers do not have the WHEA capability.
- Each of the two DRAM memory controllers has error injection on write/read data paths including address/source match. See Section 6.16.2, "Error Injection Control and Registers — Memory Controller".

## 6.10 SoC UnCore and I/O Error Handling

Figure 6-3, "Intel Atom® Processor C3000 Product Family Error Reporting" shows high level error handling scheme. The SoC receives PCIe error messages from downstream devices. The SoC logs these errors and other internal device errors. More details are provided in the following sections.

**Figure 6-3. Intel Atom® Processor C3000 Product Family Error Reporting**



## 6.11 Hardware Error Classification and System Events

In the SoC, hardware errors are classified as two types:

- Correctable Errors
- Uncorrectable Errors

This classification separates hardware errors that result in degraded performance from those that result in functional failures.

Uncorrectable hardware errors can be further classified as:

- Non-Fatal Error
- Fatal Error

Classifying hardware errors in this way provides the platform with ways to map a hardware error to a suitable handling mechanism. Each severity can trigger a System Event according to the GLREG Global System Event Map (GLREG_GSYSEVTMAP) register located in Host Root Configuration Space at B0;D5;F0, offset 250h. The System Event choices are either generate a Non-Maskable Interrupt (NMI), a System Management Interrupt (SMI), or generate no System Event.

The GLREG_GSYSEVTMAP register provides software the flexibility to map an Uncorrectable Error to the suitable error severity. For example, a platform design may choose to map an Uncorrectable DRAM ECC Error as a Non-Fatal Error while another platform design may require mapping the same error as a Fatal Error.

The software/firmware can choose to alter the default mapping after power-on.

### 6.11.1 Correctable Errors

At power-on, the GLREG_GSYSEVTMAP register is default-set to map Correctable Errors to an SMI System Event.

Hardware correctable errors include those error conditions where the system recovers without any loss of information. The hardware corrects these errors, and no software intervention is required. For example, a link CRC error which is corrected by the data link level retry is considered a correctable error.

- An error is corrected by the hardware without software intervention. System operation is degraded, but its functionality is not compromised
- A correctable error is logged and reported in a implementation specific manner:
  - Upon the immediate detection of the correctable error
  - Upon the accumulation of errors reaching to a threshold

### 6.11.2 Uncorrectable Errors

At power-on, the GLREG_GSYSEVTMAP register is default-set to map Non-Fatal Uncorrectable Correctable Errors to an SMI System Event.

An uncorrected error is a hardware error condition that cannot be corrected by the hardware or by the firmware. Uncorrected errors are either fatal or non-fatal.

#### 6.11.2.1 Fatal Errors

A fatal hardware error is an uncorrected or uncontained error condition that is determined to be unrecoverable by the hardware. When a fatal uncorrected error occurs, the system is restarted to prevent propagation of the error.

#### 6.11.2.2 Non-Fatal Errors

A non-fatal hardware error is an uncorrected error condition from which the Operating System or firmware can attempt recovery by trying to correct the error. These are also referred to as correctable or recoverable errors.

Isolating non-fatal from fatal errors provides system management software the opportunity to recover from the error without reset and disturbing other transactions in progress. Devices not associated with the transaction in error are not impacted by the error. An example of recoverable error is an ECC uncorrectable error that affects only the data portion of a transaction:

- The error is not corrected by the hardware and requires software intervention for correction
- The error is not corrected. Data integrity is compromised, but system operation is not compromised
- This error requires immediate logging and reporting of the error to the CPU

### 6.11.3 Software Correctable Errors

Software correctable errors are considered as a recoverable error. These errors include those error conditions where the system recovers without any loss of information. Software intervention is required to correct these errors:

- This error requires immediate logging and reporting of the error to the CPU
- The firmware or other system software layers take corrective actions

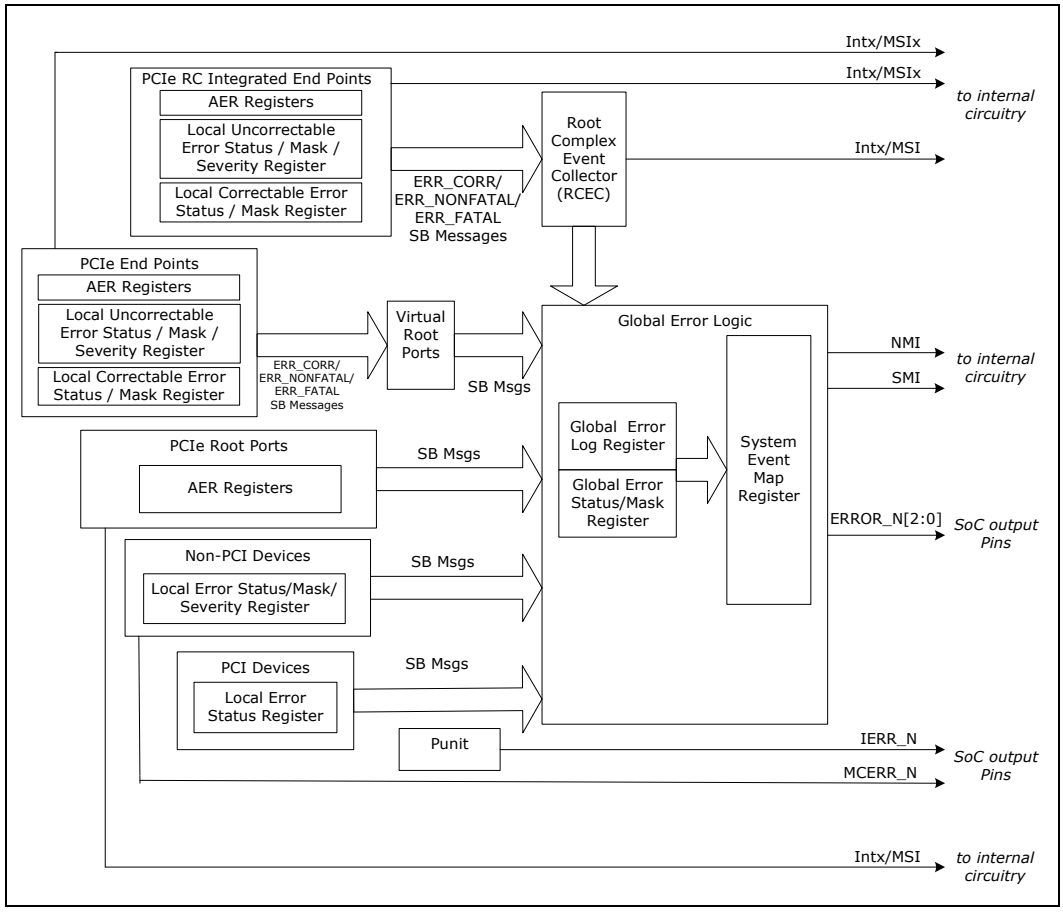
Data integrity is not compromised with such errors.

## 6.12 SoC UnCore and I/O Error Reporting

The SoC logs and reports the detected errors via "system event" generations. In the context of error reporting, a system event is an event that notifies the system of the error. Two types of system events can be generated -- an NMI or SMI to the SoC, or an ERROR_N[2:0] pins indication to the platform. The SoC responds to a system event (NMI or SMI) and takes the appropriate action to handle the error. An external agent such as BMC may monitor ERROR_N[2:0] pins to determine the health of SoC and interrupt the SoC. In some severe error cases, if the SoC is no longer responding to system events resulting from an error(s), the ERROR_N[2:0], IERR_N, and MCERR_N pins provide a way to notify an external agent of the error. The external agent can then perform a reset to recover the Denverton SoC functionality.

The Figure 6-4 illustrates internal and external error reporting. The SoC receives PCIe error messages from downstream devices. The SoC logs these errors and other internal errors.

**Figure 6-4. Error Handling Architecture**



The SoC detects errors from the PCIe* links and the SoC internal device errors. The errors are first logged and mapped to an error severity, and then mapped to a system event(s) for error reporting.

SoC error-reporting features are summarized below. Details are in the following sections:

- Detects and logs PCIe links and SoC internal device errors
- First error/next error detection and logging for correctable/uncorrectable local errors
- Allows flexible mapping of local uncorrectable errors to fatal or non-fatal error classes
- First/next error detection and logging for correctable, non-fatal, and fatal global errors
- Flexible error reporting using multiple reporting mechanisms
- Supports PCIe error reporting mechanism based on the Root Complex Event Collector (RCEC)

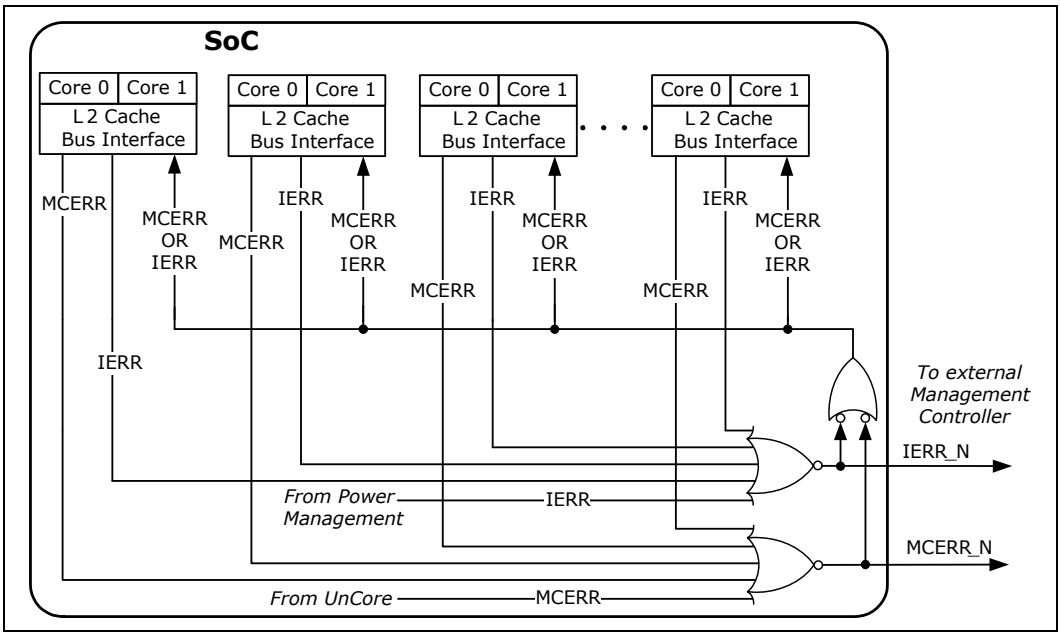
The SoC provides direct mapping of system errors to NMI or SMI. The System Error (SERR) mechanism is not used to do this.

### 6.12.1 Reporting Error to an External Device

Detected errors can be forwarded to an external device (e.g., a BMC) using the SoC ERROR_N[2:0] signal pins.

The SoC escalates machine check errors (MCERR_B) and unrecoverable internal errors (IERR) to the external BMC and to the CPU cores for handling non-maskable and other fatal errors in the system. See Figure 6-5. The MCERR_N and IERR_N signals are broadcast to the CPU and to the external platform.

**Figure 6-5. MCERR and IERR Handling**




The MCERR_N signal signifies a catastrophic internal error, a condition which requires immediate attention or possibly shutdown. When this error occurs, the processor core may not be able to execute reliably through the INT18 handler. The following are some possible cases of such catastrophic errors:

- Retirement watchdog time-out from the core
- Internal error detected by the SoC power management circuitry

The MCERR_N signal signifies a machine check error occurred and that SoC Machine Check Architecture registers, accessible through the MSRRD and MSRWR instructions, may have additional information concerning the error.

*Note:* Board designs must not consider IERR_N and MCERR_N valid until after the PMU_PLTRST_N (Platform Reset) signal is deasserted by the SoC. When the SoC is powered-up or a cold boot, the IERR_N and MCERR_N signals may be unstable and falsely signal an internal error or machine check error before the platform reset is deasserted by the SoC.

### 6.12.2 PCI Express INTx and MSI

PCIe INTx and MSI are supported via PCIe standard error reporting. The SoC forwards MSI generated from downstream PCIe devices to the CPU. Also, PCIe Root ports and RCEC in the SoC generate Intx/MSI interrupts for error reporting if enabled. See PCIe specification for details on PCIe standard and advanced error reporting capability.

## 6.13 Overview of SoC Error Registers

The SoC contains sets of error registers to support error reporting. These error registers are assumed to be sticky unless specified otherwise. Sticky means the values of the registers are retained even after a hard reset — they can only be cleared by software or by power-on reset.

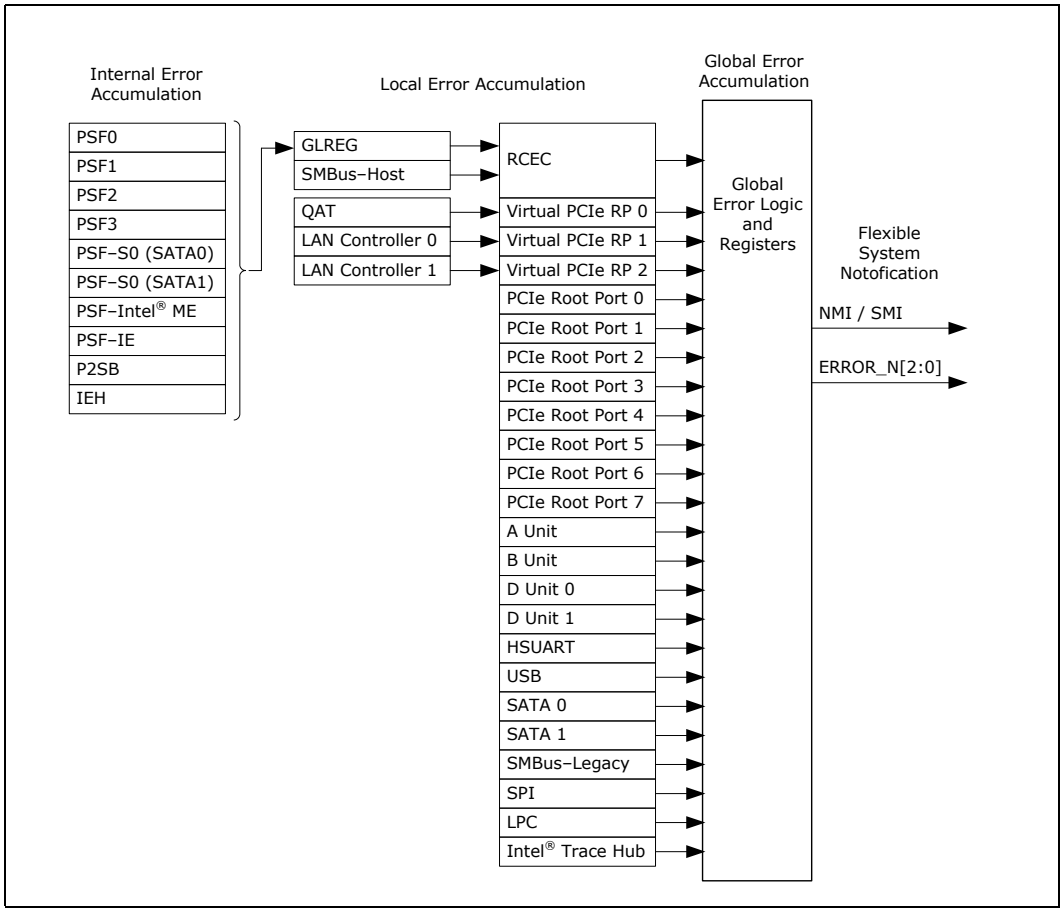
There are two levels of hierarchy for the SoC error registers:

- Local Error Registers
- Global Error Registers

See Figure 6-6, "Local and Global Error Register Structure". The Local Error Registers are associated with the local, integrated devices. The Global Error Registers collect the errors reported by the Local Error Registers and map them to system events and/or to SoC output signal pins.

*Note:* The PSF, IEH, P2SB, A, B, C, and D Units are physical functions within the SoC. They are part of the internal fabric that handles I/O transactions to/from the CPU cores and the system memory subsystem. This document does not provide details of these internal functions other than errors they detect and report.

**Figure 6-6. Local and Global Error Register Structure**



### 6.13.1 Local Error Registers

Table 6-5, "Local Error Register General Descriptions" provides a general description of the SoC Local Error registers. Not all SoC error-detecting mechanisms implement all of these registers. Some have additional Local Error registers not listed here.

**Table 6-5. Local Error Register General Descriptions**

| Local Error Register Name | Local Error Register Description |
|---|---|
| **Uncorrectable Errors** | |
| Uncorrectable Error Status (not AER) | When a specific Uncorrectable Error associated with an SoC local interface occurs, its corresponding bit in this error status register is set. Each error can be individually masked by the Uncorrectable Error Mask register. A masked error bit is not set for any subsequent detected error and is not recorded or reported in the Uncorrectable Header Log register and does not update the Uncorrectable FERR/NERR registers. |
| Uncorrectable Error Mask (not AER) | |
| Uncorrectable Error Status (AER) | The Uncorrectable Error Status register reports error status of individual correctable error sources on a PCI Express device Function. Software may clear an error status by writing a 1 to the respective error bit. When the respective bit is set in the Uncorrectable Error Mask, the error is set in the Uncorrectable Error Status register, but it is not reported to the PCI Express Root Complex via a PCI Express Error Signaling Message, and the Header Log and First Error Pointer registers are unmodified. |
| Uncorrectable Error Mask (AER) | |
| Uncorrectable Error Severity | Controls whether an individual error is reported as a Non-Fatal (register = 0) or Fatal (register = 1) error. |
| First Uncorrectable Error | The First Uncorrectable Error Register (FERR) logs the first occurrence of an error, while the Next Uncorrectable Error Register (NERR) logs the subsequent occurrence of the errors. The Uncorrectable FERR and NERR do not log errors that are masked. The Uncorrectable FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding Uncorrectable FERR error bit in the Uncorrectable Error Status Register by software. |
| Next Uncorrectable Error | |
| Uncorrectable Header Log (not AER) | Logs the header of the first Uncorrectable Error associated with the particular local interface. |
| **Correctable Errors** | |
| Correctable Error Status (not AER) | When a specific Correctable Error associated with an SoC local interface occurs, its corresponding bit in this error status register is set. Each error can be individually masked by the Correctable Error Mask register. A masked error bit is not set for any subsequent detected error and is not recorded or reported in the Uncorrectable Header Log register and does not update the Correctable FERR/NERR registers. |
| Correctable Error Mask (not AER) | |
| Correctable Error Status (AER) | The Correctable Error Status register reports error status of individual correctable error sources on a PCI Express device Function. Software may clear an error status by writing a 1 to the respective error bit. When the respective bit is set in the Correctable Error Mask, the error is set in the Correctable Error Status register, but it is not reported to the PCI Express Root Complex via a PCI Express Error Signaling Message, and the Header Log and First Error Pointer registers are unmodified. |
| Correctable Error Mask (AER) | |
| First Correctable Error | The First Correctable Error Register (FERR) logs the first occurrence of an error, while the Next Correctable Error Register (NERR) logs the subsequent occurrence of the errors. The Correctable FERR and NERR do not log errors that are masked. The Correctable FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding Correctable FERR error bit in the Correctable Error Status Register by software. |
| Next Correctable Error | |
| Correctable Header Log (not AER) | Logs the header of the first Correctable Error associated with the particular local interface. |
| **AER Specific** | |
| Advanced Error Capabilities and Control (AER) | Identifies AER capabilities and the bit position of the first error reported in the (AER) Uncorrectable Error Status register. |
| Advanced Error Header Log (AER) | Header (four DWords) of the Transaction-Level Packet (TLP) associated with the error. |

### 6.13.2 SoC Global Error Registers

The SoC provides Global Error Registers and circuitry. The Global Error Registers are not defined in any industry specification. The provide a quick way for software to start searching for the error that caused the system error-reporting event.

**Table 6-6. List of SoC Global Error Registers**

| Host Root Configuration Space B0:D4:F0 Offset (hex) | Global Error Register | Register Name |
|---|---|---|
| 200 | GLREG_GCOERRSTS | Global Correctable Error Status |
| 204 | GLREG_GNFERRSTS | Global Non-Fatal Error Status |
| 208 | GLREG_GFAERRSTS | Global Fatal Error Status |
| 20C | GLREG_GERRMSK | Global Error Mask |
| 228 | GLREG_GTIME | Global Error Timer |
| 230 | GLREG_GCOFERRTIME | Global Correctable FERR Error Time Stamp |
| 238 | GLREG_GNFFERRTIME | Global Nonfatal FERR Error Time Stamp |
| 240 | GLREG_GFAFERRTIME | Global Fatal FERR Error Time Stamp |
| 248 | GLREG_GSYSEVTSTS | Global System Event Status |
| 24C | GLREG_GSYSEVTMSK | Global System Event Mask |
| 250 | GLREG_GSYSEVTMAP | Global System Event Map |
| 254 | GLREG_ERRPINCTRL | Error Pin Control |
| 258 | GLREG_ERRPINSTS | Error Pin Status |
| 25C | GLREG_ERRPINDATA | Error Pin Data |

### 6.13.3 First Error and Next Error Log Registers

This section pertains to both Local Error logging and Global Error logging. The error-log registers names contain FERR (First Error) and NERR (Next Error). First Errors and Next Errors are captured at both the local level (correctable and uncorrectable) and Global level (correctable, non-fatal and fatal). PCI Express specifies its own error-logging mechanism which is not be described here and can be found in the *PCI Express Base Specification Revision 3.0*.

For Global Error logging, the SoC categorizes a detected error based on the error severity:

- Fatal Error
- Non-Fatal Error
- Correctable Error

Each category includes two sets of error logging:

- First Error register (FERR)
- Next Error register (NERR)

The FERR register stores the information associated with the first detected error, while NERR stores the information associated with the subsequent detected errors after the first error. Both FERR and NERR log the error status in the same format. They indicate errors that can be detected by the SoC with one bit assigned to each type of error. The first error event is indicated by setting the corresponding bit in the FERR status register; a subsequent error is indicated by setting the corresponding bit in the NERR register. In addition, the local FERR registers logs the header of the erroneous cycle. Both First Error and Next Error trigger system events.

Once the First Error and the Next Error have been indicated and logged, the log registers for that error remain valid until either

The first error bit is clear in the associated error status register, or

A reset after loss of power.

Software clears an error bit by writing 1 to the corresponding bit position in the error status register.

The SoC uses this procedure tor updating the FERR and NERR registers and error logs:

1. First Error event is indicated by setting the corresponding bit in the FERR status register, a subsequent error is indicated by setting the corresponding bit in the NERR status register.
2. If the same error occurs before the FERR status register bit is cleared, it is not logged in the NERR status register.
3. In the case of simultaneous multiple errors with same severity, any two errors may be logged in FERR and NERR registers.
4. Updates to error status and error log registers appear atomic to the software.
5. Once the information of the first error is logged in the FERR log register, the logging of FERR log registers is disabled until the corresponding FERR error status is cleared by the software.
6. Error status registers, Error Mask registers and Error Log registers are cleared by the power-on reset only. The contents of Error Log registers are preserved across a reset (while power-good is sensed).

### 6.13.4 General Local and Global Error-Register Flow

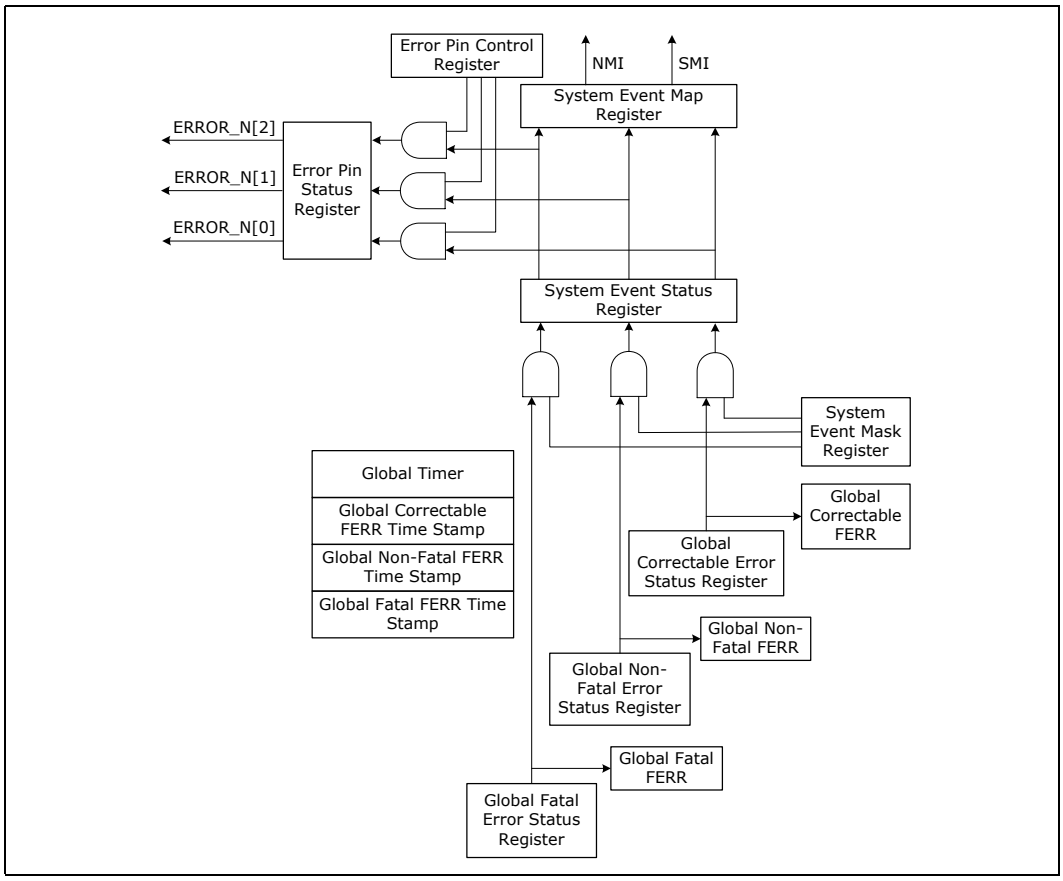
1. Upon a detection of a Local error, the corresponding Local error status is set if the error is unmasked. Otherwise the error bit is not set and the error is not propagated upstream towards the SoC Root Complex.
2. The Local uncorrectable error is mapped to its associated error severity defined by the Uncorrectable Error Severity register. Setting of the Local Error Status register bit causes the logging of the error. Fatal and Non-fatal errors are logged in the Local Uncorrectable FERR and NERR registers, while correctable errors are logged in the Local Correctable FERR and NERR registers. PCI Express errors are logged according to the PCI Express specification.
3. The local FERR and NERR logging events are forwarded to the Global FERR and NERR registers. The report of Local FERR/NERR sets the corresponding Global Error Status bit if the particular global error is unmasked. Otherwise the global error bit is not set and the error is propagated. As mentioned in Section 6.13.3, "First Error and Next Error Log Registers", the Global FERR logs the first occurrence of local FERR event in the SoC while the Global NERR logs the subsequent local NERR event.
4. Correctable error is logged in the Global Correctable FERR/NERR registers, Non-Fatal is logged in the Global Non-Fatal FERR/NERR registers, and Fatal error is logged in the Global Fatal FERR/NERR registers.
5. The Global Error Status register reports the error with its associated error severity to the Global System Event Status (GLREG_GSYSEVTSTS) register. The system event status is generated if the system event reporting is unmasked for the error severity in the Global System Event Mask (GLREG_GSYSEVTMSK) register. Otherwise the bit is not set and the error is not reported.
6. Setting of the system event bit triggers a system event generation according the mapping defined in the Global System Event Map (GLREG_GSYSEVTMAP) register. The associated system event is generated for the error severity and the error indication dispatched to the CPU/BMC (interrupt for CPU or Error Pin for BMC).
7. The global log and local log registers provide the information to identify the source of the error. Software can read the log registers and clear the global and local error status bits.
8. Since the error status bits are edge triggered, a 0-to-1 transition is required to set the bit again. While the error status bit (local, global, or system event) is set to 1, all incoming error reporting to the respective error status register is ignored (no 0-to-1 transition).
    a. When a write occurs to clear the Local Error Status bit, the local error register circuitry re-evaluates the "OR" output of its error bits and reports it to the Global Error register. However, if the Global Error Status bit is already set, then the report is ignored.
    b. When write occurs to clear error status bit, the Global Error register circuitry re-evaluates the "OR output" of its error bits and reports it to the Global System Event Status (GLREG_GSYSEVTSTS) register. However, if the system event status bit is already set, then the report is not generated.
    c. Software can optionally mask or unmask the system event generation (interrupt or error pin) for an error severity in the Global System Event Mask (GLREG_GSYSEVTMSK) register while clearing the Local and Global error registers.

9. Software has the following options for clearing error status registers:
   a. Read the Global and Local Error Log registers to identify the source of the error. Clear local error bits. This does not cause the generation of an interrupt with the global bit still set. Then clear the global error bit and write to the local error register again with all 0s. Writing 0s to the local status does not clear any status bits, but will cause the re-evaluation of the error status bits. An error is reported if there are any uncleared local error bits.
   b. Read the Global and Local Error Log registers to identify the source of the error and mask the error reporting for the error severity. Clear the system event and global error status bits. This causes the setting of the system event status bit if there are other global bits still set. Then clear the local error status bits. This causes the setting of the Global Error Status bit if there are other local error bits still set. Then unmask the system event to cause the SoC to report the error.
10. The FERR register logs the information of the first error detected by the associated error status register (Local or Global). The FERR log remains unchanged until all bits in the respective error status register are cleared by the software. When all error bits are cleared, FERR logging is re-enabled.

## 6.13.5 Global System Event Reporting

Provided that the Global System Event Mask (GLREG_GSYSEVTMSK) register is not set to prevent the generation of system events, the Global System Event Map (GLREG_GSYSEVTMAP) register determines which system event is generated to the CPU for a given Global Error severity. See Figure 6-7, "Global Error Registers and System Event Reporting".

**Figure 6-7. Global Error Registers and System Event Reporting**



Each severity (Fatal, Non-Fatal, Correctable) is programmable to generate either an NMI, SMI or not generate a system event. The default settings are:

- Fatal Error - Generate an NMI
- Non-Fatal Error - Generate an SMI
- Correctable Error - Generate an SMI

The Global System Event can also be generated to the platform board through the SoC output signal pins which are shown in Table 6-1, "Signal Names and Descriptions".

- ERROR_N[2] - Fatal Error
- ERROR_N[1] - Non-Fatal Error
- ERROR_N[0] - Correctable Error

The programmable Error Pin Control (GLREG_ERRPINCTRL) register determines how these error pins function. Each pin can also be disabled by this register. The signal pin can be configured to assert when a Global Error of its associated severity is set in the Global System Event Status (GLREG_GSYSEVTSTS) register. It can also be configured to operate as a programmed-controlled signal similar to a general-purpose I/O signal.

The Error Pin Data (GLREG_ERRPINDATA) register provides the data value when the error pin is configured as a program-controlled output signal. The assertion and deassertion of each of the three ERROR_N pins can be programmed. The particular error-pin signal follows the value programmed here. This mechanism only functions when 2-bit Error Enable field of the Error Pin Control (GLREG_ERRPINCTRL) register is set to 01 for the particular Error Pin Data bit.

ERROR_N[1] and ERROR_N[0] are default set to operate in this program-controlled mode. ERROR_N[0] is default-set to assert the pin's signal when a Fatal error is set in the Global System Event Status (GLREG_GSYSEVTSTS) register.

## 6.13.6 Error Counters

The Error Counters allow the system management controller to monitor the component's health by periodically reporting the Correctable Error Count. The error RAS structure already provides a First Error status and a Second Error status. Because the response time of system management is on the order of milliseconds, it is not possible to read and clear the error logs in time to detect short bursts of errors across the component. Over a long period of time the software uses these values to monitor the rate of change in error occurrences. This helps identify potential component degradations, especially with respect to the system memory interface.

### 6.13.6.1 Error Counter Requirements

A register with one-hot encoding selects the error types that participate in error counting. It is unlikely that more than one error will occur within a cluster at a given time. Therefore, it is not necessary to count more than one occurrence in one clock cycle. The selection register ORs together all of the selected error types to form a single count enable. This means that only one increment of the counter occurs for one or all types selected. Register attributes are set to write-1-to-clear.

Each of these clusters has a set of error counter/control registers.

- The IEH device (B0:D4:F0) contains a seven-bit Error Counter (ERRCNT[6:0]) in the Uncorrectable Error Counter (GLREG_LUNCERRCNT) register located in Host Root Configuration Space, offset 294h.
  - GLREG_LUNCERRCNT bit 7 (ERR_OVF) is an overflow bit. All eight bits of the register are sticky with a write logic 1 to clear.
- Each of the two D-Units of the SoC System Agent contains an eight-bit Error Counter (UNCCNT[7:0]) field in its Uncorrectable Error Counter (DERRCNT) register located in Host Root Memory Space at SBREG_BAR plus offset 10_0130h (D-Unit 0) and 12_0130h (D-Unit 1).
  - UNCCNT bit 7 is an overflow bit. All eight bits of the UNCCNT field of the register are sticky with a write logic 1 to clear.
- The B-Unit of the System Agent contains a 15-bit Error Counter (ERR_CNT[14:0]) in the Uncorrectable Error Count (B_CR_UNCERRCNT_0_0_0_MCHBAR) register located in Host Root Memory Space at MCHBAR plus offset 6E1Ch.
  - B_CR_UNCERRCNT_0_0_0_MCHBAR register bit 15 (OVERFLOW) is an overflow bit. Bits [15:0] bits of the register are sticky with a write logic 1 to clear.
- The A-Unit of the System Agent contains a 15-bit Error Counter (ERR_CNT[14:0]) in the Uncorrectable Error Count (A_CR_UNCERRCNT_0_0_0_MCHBAR) register located in Host Root Memory Space at MCHBAR plus offset 65A0h.
  - A_CR_UNCERRCNT_0_0_0_MCHBAR register bit 15 (OVERFLOW) is an overflow bit. Bits [15:0] bits of the register are sticky with a write logic 1 to clear.
- The A-Unit of the System Agent contains a 15-bit Error Counter (ERR_CNT[14:0]) in the Correctable Error Count (A_CR_CORERRCNT_0_0_0_MCHBAR) register located in Host Root Memory Space at MCHBAR plus offset 65B8h.
  - A_CR_CORERRCNT_0_0_0_MCHBAR register bit 15 (OVERFLOW) is an overflow bit. Bits [15:0] bits of the register are sticky with a write logic 1 to clear.

## 6.14 Error Reporting Flow — PCI Express Devices/Functions

- PCIe Root Ports 0 through 7
- Virtual PCIe Root Ports 0 through 2
- PCIe Endpoints behind the Virtual Root Ports
  - — Intel® QuickAssist Technology agent
  - — LAN Controller 0 and 1
- PCIe Root Complex Integrated Endpoints via RCEC
  - — SMBus - Host Controller
  - — Integrated Error Handler (IEH) agent

The PCI Express* architecture provides three complementary mechanisms which allow the agent detecting an error to alert the system or another device that an error has occurred. The three mechanisms are:

- Through the Completion Status of a transaction
- In-band Error-Signaling Messages
- Error Forwarding (also known as data poisoning)

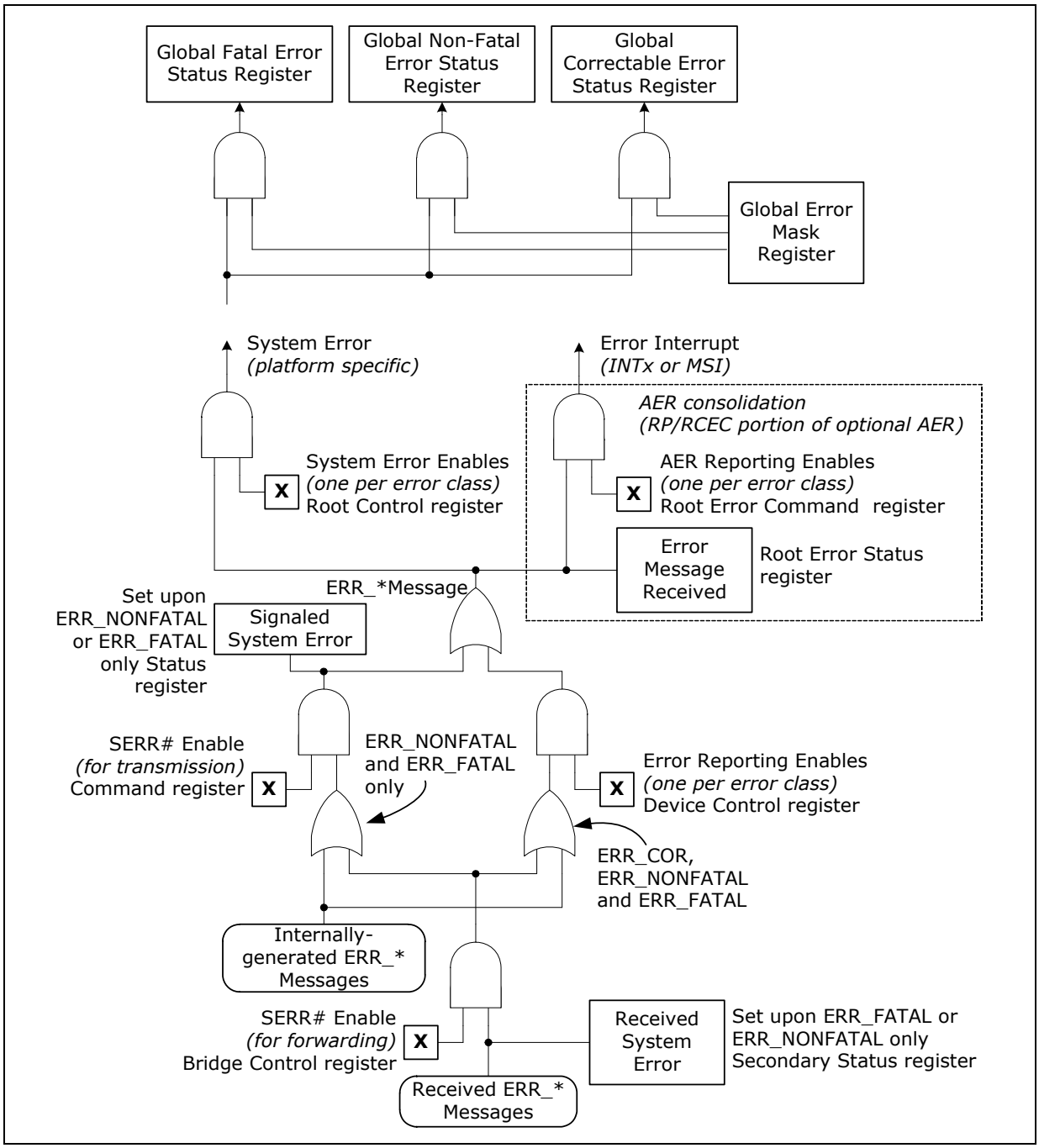
Section 6.2.7 “Error Listing and Rules” of the *PCI Express Base Specification Revision 3.0* lists the actions that an error-detecting agent is expected to take to report and log an error using the above mechanisms. Section 2.2.8.3 “Error Signaling Messages” of the same specification defines the Error Signaling Messages that are used to signal errors that occur on specific PCI Express transactions and errors that are not necessarily associated with a particular transaction. These Messages are initiated by the agent that detected the error. The three Error Signaling Messages are:

- ERR_COR - The device/function detected a Correctable Error.
- ERR_NONFATAL - The device/function detected a Non-Fatal Error.
- ERR_FATAL - The device/function detected a Fatal Error.

### 6.14.1 Upstream Flow of Error Signal Messages

The upstream flow of the agents' Error Signal Messages to the SoC Global Error Registers is shown in Figure 6-8, "PCIe Error Signaling and Logging". PCI Express endpoint devices attached to the SoC Root Ports send these to the Root Ports where they are channeled along with the Root Port internal errors to the SoC Global Error Registers.

**Figure 6-8. PCIe Error Signaling and Logging**



### 6.14.2 How a PCIe Device Determines Which Error Message to Send

Section 6.2 "Error Signaling and Logging" of the *PCI Express Base Specification Revision 3.0* describes the PCIe In-band Error-Signaling Messages. It also provides a flowchart showing how the PCIe device/function determines which error Message, if any, to send to the SoC Root Complex.

All of the flow diagrams provided above show how the Advanced Error Reporting Capability (AER), an optional PCI Express Extended Capability, affects the flow. All of the PCI Express Devices/Functions mentioned in this sub-section have the AER Capability.

Besides the AER Capability, each PCIe device/function has a Device Status register that is part of its PCI Express Capability structure and contains indications of detected errors. Each PCIe device/function also has a Device Control Register that is part of its PCI Express Capability structure. This register contains bits that, in conjunction with other bits, provides software the capability to mask (prevent) the ERR_COR, ERR_NONFATAL, and ERR_FATAL error signaling messages from being transmitted towards the Root Complex.

If an individual error is masked when it is detected, its error status bit is still affected, but no error reporting Message is sent to the Root Complex, and the Header Log and First Error Pointer registers are unmodified. See Section 6.2.3.2.2 "Masking Individual Errors" and the flowchart in Section 6.2.5. "Sequence of Device Error Signaling and Logging Operations" in the *PCI Express Base Specification Revision 3.0*.

The Root Complex internally translates these error indications into platform-level events. This includes setting a possible NMI or SMI (interrupt), and ERROR_N[2:0] output signal pins.

Since they are part of the SoC Root Complex, the Root Ports, Virtual Root Ports, and RCEC interface directly to the SoC Global Registers rather than use the PCIe Error-Signaling Messages mentioned above. Each determines the error severity based on what is programmed in its PCIe Root Control (ROOTCTL) register and, as an option, sends the error to the SoC Global Registers.

As a PCIe Root Complex Integrated Endpoint (RCiE), the SMBus-Host Controller (B0:D18:F0) uses the SoC Root Complex Event Collector (RCEC) to pass errors to the Global Error logic and registers. As part of its AER Capability structure, the RCECs Root Error Status (RCEC_ROOTERRSTS) register and the Error Source Identification (RCEC_ERRSRCID) register that are used to capture the SMBus-Host Controller errors. They are located in Host Root Configuration Space at B0:D5:F0, offsets 130h and 134h respectively. For this to all happen the error signaling messages must be sent to the RCEC and not blocked by the SMBus-Host's Device Control Register that is part of its PCI Express Capability structure. RCEC also has the AER Root Error Command register (RCEC_ROOTERRCMD) at offset 12Ch that can be used to prevent error signaling messages from going to the Root Complex and the Global Error logic and registers.

Also as an RCiE, the SoC Integrated Error Handler (IEH) agent (B0:D4:F0) accumulates internal fabric hardware errors, all considered uncorrectable errors, and report them to the RCEC to pass-on to the Global Error logic/registers and generate an NMI or SMI, and optionally assert the SoC ERROR_N[2:0] output signal pins. This is done through Uncorrectable Internal Error (UIE), bit 22 of the RCEC Uncorrectable Error Status (RCEC_ERRUNCSTS) register located in Host Root Configuration Space at B0:D5:F0, offset 104h.

According to the *PCI Express Base Specification Revision 3.0*, the only method of recovering from an Uncorrectable Internal Error is reset or hardware replacement.

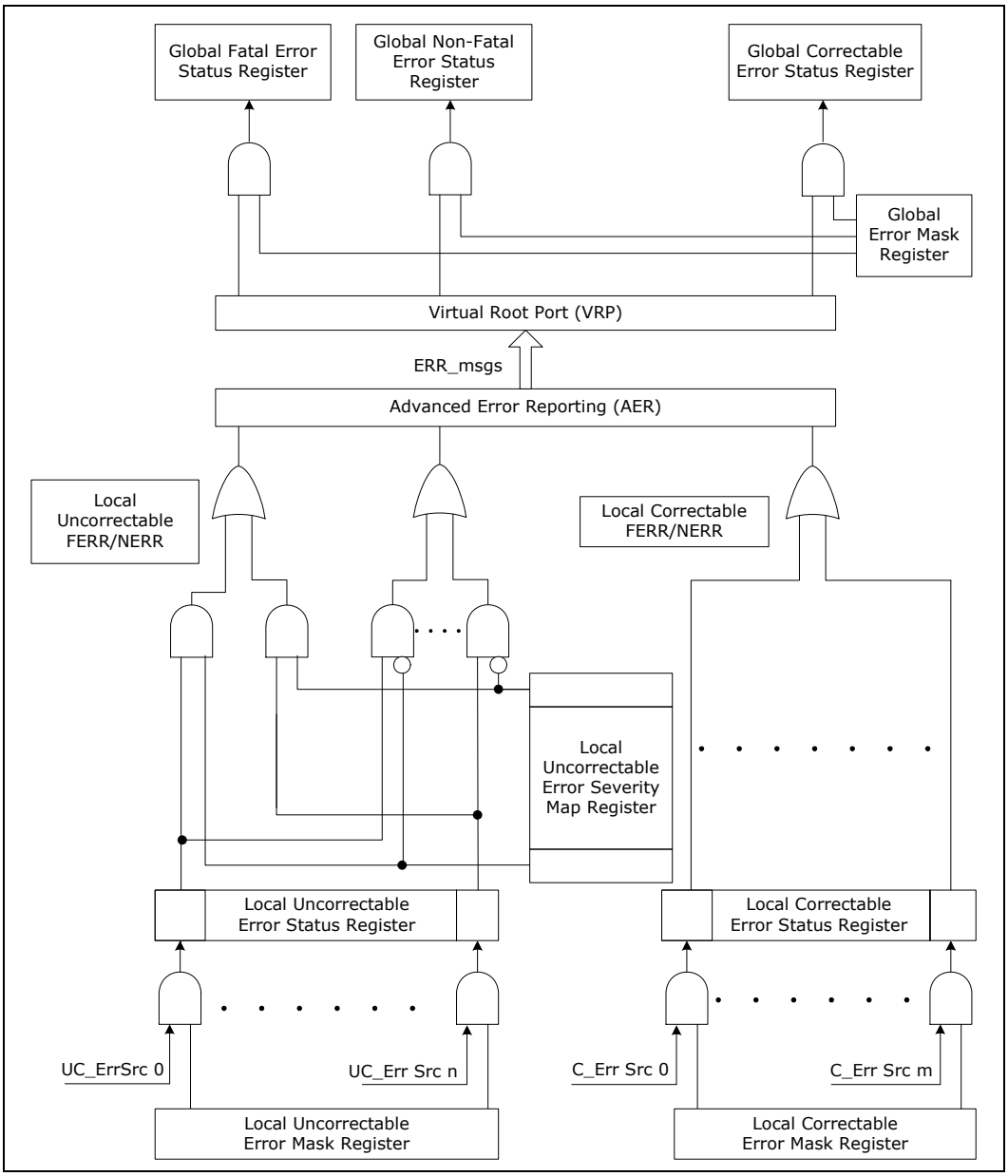
Like RCEC for the SMBus-Host mentioned above, each of the SoC PCIe Root Ports and Virtual Root Ports, as part of its AER Capability structure, has a Root Error Status (ROOTERRSTS) register and an Error Source Identification (ERRSRCID) register to capture errors signaled to them and the transaction Requester ID. Each also has a Root Error Command (ROOTERRCMD) that can be used to prevent error-signaling messages from propagating to the Root Complex and the Global Error logic and registers.

*Note:* SoC Machine Check events do not occur from hardware errors detected by PCIe agents.

## 6.14.3 PCIe Endpoint Devices Behind Virtual Root Port Error Handling

The integrated Intel® QuickAssist Technology controller and the integrated LAN Controllers are PCI Express endpoint devices. Each is attached to a dedicated "Virtual" Root Port (VRP). Each has Local Error Registers and an Error Signaling Message flow to the SoC Global Error Registers that is shown in Figure 6-9, "Error Signaling and Logging for PCIe Root Complex Integrated Endpoints."

**Figure 6-9. Error Signaling and Logging for PCIe Root Complex Integrated Endpoints**




These PCIe endpoint devices support AER registers for logging and reporting internal fabric and device specific errors. Device-specific errors are logged into Local Error registers and reported using Uncorrectable Internal Error Status (Bit 22) of ERRUNCSTS register and Corrected Internal Status (Bit 14) of the ERRUNCSTS register.

### 6.14.4 PCI Express Root Ports

While PCIe Root Port error reporting is described in the *PCI Express Base Specification Revision 3.0*, the following paragraphs highlight important aspects of PCIe Root port error reporting mechanism.

#### 6.14.4.1 Unsupported Transactions and Unexpected Completions

If a SoC Root Port receives a legal PCIe-defined packet that is not included in PCIe supported transactions, the SoC treats that packet as an unsupported transaction and follows the PCIe rules for handling Unsupported Requests (UR). If the Root Port receives a Completion with a Requester ID set as the Root Port's Requester ID and there is no matching request outstanding, then this is considered an Unexpected Completion (UC). Also, the SoC Root Ports detects malformed packets from the PCI Express interface and reports them as errors per the PCI Express specification.

If a SoC Root Port receives a Type 0 Intel Vendor-Defined message that terminates at the SoC Root Complex and does not recognize it as a valid Intel-supported message, the message is handled by the SoC as an Unsupported Request (UR) with the appropriate error escalation defined in The PCI Express specification. For Type 1 Vendor-Defined messages which terminate at the Root Complex, the SoC simply discards the message with no further action.

#### 6.14.4.2 Unconnected Ports

If a CPU transaction targets a PCIe link that is not connected to any device or the link is down (DL_Down status is received by the Root Port), the SoC treats this as a Master Abort situation. This is required for PCI bus scans to non-existent devices to go through without creating any other side effects. If the transaction is non-posted, The SoC returns all F's. Note that accesses by CPU to the Root Port registers corresponding to an unconnected PCIe interface does not generate an error.

### 6.14.5 Internal Errors — SMBus-Host Controller

The SMBus-Host Controller report its internal errors in the SMBus-Host Error Status (ERRSTS) register located in Host Root Memory Space located at SMTBAR, offset 18h.

The ERRAERMSK register located at SMTBAR, offset 14h, can be programmed to prevent individual SMBus-Host internal errors (ERRSTS) from also setting the Uncorrectable Internal Error (UIE) status, bit 22 of the SMBus-Host's AER Uncorrectable Error Status (ERRUNCSTS) register located in Host Root Configuration Space at B0:D18:F0, offset 104h. Whether or not the UIE error is signaled to the RCEC and Global Error logic depends on the setting of Uncorrectable Internal Error Mask (UIEM), bit 22 of the SMBus-Host's AER ERRUNCMSK register located at offset 108h.

## 6.14.6 Local Error Registers

The following tables list the registers associated with the error control, status, and reporting flow for the PCIe devices/functions of the SoC:

- Table 6-7, "Local Error Registers — PCI Express Root Ports"
- Table 6-8, "Local Error Registers — PCI Express RCEC and RCiEP Functions"
- Table 6-9, "Local Error Registers — PCI Express Functions Attached to Virtual Root Ports"

Register details are in the register chapters of this document. Also, reference the *PCI Express Base Specification Revision 3.0* for error register architectural details.

**Table 6-7. Local Error Registers — PCI Express Root Ports**

| Location in Host Root Configuration Space | PCIe RP0 - RP7 | Virtual PCIe RP0 - RP2 | Register Name |
|---|---|---|---|
| PCI Header Type 1 | PCICMD | PCICMD | Command |
| | PCISTS | PCISTS | Status |
| | SECSTS | SECSTS | Secondary status |
| | BCTL | BCTL | Bridge Control |
| PCIe Capability | DEVCTL | DEVCTL | Device Control |
| | DEVSTS | DEVSTS | Device Status |
| | ROOTCTL | ROOTCTL | Root Control |
| AER Capability | ERRUNCSTS | ERRUNCSTS | Uncorrectable Error Status |
| | ERRCORSTS | ERRCORSTS | Correctable Error Status |
| | AERCAPCTL | AERCAPCTL | AER Capabilities & Control |
| | AERHDRLOG1 | AERHDRLOG1 | Header Log (one entry) |
| | AERHDRLOG2 | AERHDRLOG2 | |
| | AERHDRLOG3 | AERHDRLOG3 | |
| | AERHDRLOG4 | AERHDRLOG4 | |
| | ROOTERRCMD | ROOTERRCMD | Root Error Command |
| | ROOTERRSTS | ROOTERRSTS | Root Error Status |
| | ERRSRCID | ERRSRCID | Error Source Identification |

### Table 6-8. Local Error Registers — PCI Express RCEC and RCiEP Functions

| Location in Host Root Configuration Space | RCEC | SMBus-Host | Integrated Error Handler | Register Name |
|---|---|---|---|---|
| PCI Header Type 0 | RCEC_PCICMD | PCICMD | GLREG_PCICMD | Command |
| | RCEC_PCISTS | PCISTS | GLREG_PCISTS | Status |
| PCIe Capability | RCEC_DEVCTL | DEVCTL | GLREG_DEVCTL | Device Control |
| | RCEC_DEVSTS | DEVSTS | RCEC_DEVSTS | Device Status |
| | RCEC_ROOTCTL | | | Root Control |
| AER Capability | RCEC_ERRUNCSTS | AERCAPHDR | | Uncorrectable Error Status |
| | RCEC_ERRCORSTS | ERRUNCSEV | | Correctable Error Status |
| | RCEC_ERRCORMSK | ERRCORSTS | | Correctable Error Mask |
| | RCEC_AERCAPCTL | ERRCORMSK | | AER Capabilities & Control |
| | RCEC_AERHDRLOG1 | AERCAPCTL | | Header Log (one entry) |
| | RCEC_AERHDRLOG2 | AERHDRLOG1 | | |
| | RCEC_AERHDRLOG3 | AERHDRLOG2 | | |
| | RCEC_AERHDRLOG4 | AERHDRLOG3 | | |
| | RCEC_ROOTERRCMD | AERHDRLOG4 | | Root Error Command |
| | RCEC_ROOTERRSTS | | | Root Error Status |
| | RCEC_ERRSRCID | | | Error Source Identification |

### Table 6-9. Local Error Registers — PCI Express Functions Attached to Virtual Root Ports

| Location in Host Root Configuration Space | LAN Primary Function and Virtual Functions | Intel® QuickAssist Technology Primary Function | Intel® QuickAssist Technology Virtual Functions | Register Name |
|---|---|---|---|---|
| PCI Header Type 0 | CMD_STAT_REG | PPCICMD | PCICMD | Command |
| | | PPCISTS | PCISTS | Status |
| PCIe Capability | DCTL_DSTS | PPDCNTL | DEVCTL | Device Control |
| | | PPDSTAT | DEVSTS | Device Status |
| AER Capability | UEV | PPAERUCSEV | ERRUNCSEV | Uncorrectable Error Severity |
| | CES | PPAERCS | ERRCORSTS | Correctable Error Status |
| | CEM | PPAERCM | ERRCORMSK | Correctable Error Mask |
| | AECC | PPAERCAPCTL | AERCAPCTL | AER Capabilities & Control |
| | HL_DW1 | PPAERHDRLOG1 | AERHDRLOG1 | Header Log (one entry) |
| | HL_DW2 | PPAERHDRLOG2 | AERHDRLOG2 | |
| | HL_DW3 | PPAERHDRLOG3 | AERHDRLOG3 | |
| | HL_DW4 | PPAERHDRLOG4 | AERHDRLOG4 | |

## 6.15 Error Reporting Flow — Legacy PCI Devices/Functions

- SATA0 and SATA1 Controllers
- USB Combo Controller
- LPC Controller
- SMBus - Legacy Controller
- SPI Controller
- HSUART Controller
- Intel® Trace Hub

Section 2.2.5 "Error Reporting Pins" of the *PCI Local Bus Specification, Revision 3.0*, defines the Parity Error (PERR#) and System Error (SERR#) error-reporting signal pins. According to the specification, these error signals can be enabled/disabled in the PCI Command register. PERR# is only used for the reporting of data parity errors during all PCI transactions except a Special Cycle. SERR# is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. According to the specification, when an agent issues the SERR#, a non-maskable interrupt (NMI) is generated.

While there are no SoC signal pins involved, the integrated PCI devices/functions of the SoC use a similar way to report transaction errors to the SoC Root Complex, which internally translates error indications to platform-level events. An SERR# is considered an Uncorrectable, Fatal Error by the SoC. The SoC has no equivalent to PCI Parity Error PERR# signal because there is no true, physical PCI bus within the SoC. Even so, the SoC internal I/O fabric does detect and report internal parity errors to the SoC RCEC.

The Command (offset 4) and Status (offset 6) registers in device's Host Root Configuration Space play a role in how the device reports errors and the transaction errors that are detected. See Section 3.7.4 "Error Reporting" of the *PCI Local Bus Specification, Revision 3.0* for additional details.

Besides using its PCI Status register to report SERR# errors, the SATA Controllers report Fatal Errors in the General Configuration (SATAGC) register located at offset 9Ch of the SATA Configuration Space.

Besides a PCI Status register, the USB Combo Controller reports Fatal Errors in the USB Status (USBSTS) register located in Host Root Memory Space at MBAR plus offset 84h.

The LPC, SMBus - Legacy, SPI, and HSUART Controllers can each report a detected Uncorrectable Fatal Error through its PCI Device Status Register (STS) located at offset 6h for the function in Host Root Configuration Space. The Local Error Registers for the Intel® Trace Hub are not documented in this document.

The Error Status of each of the legacy PCI devices is represented in the SoC Global Error Register.

Errors detected by the SoC Legacy PCI devices are routed to the SoC Global Error logic/registers and can generate an NMI or SMI, and optionally assert the SoC ERROR_N[2:0] output signal pins. Machine Check events do not occur from these types of hardware errors.

### 6.15.1 Local Error Registers

Table 6-10, "Local Error Registers — SATA, USB, LPC and SMBus-Legacy" and Table 6-11, "Local Error Registers — SPI and HSUART" list the registers associated with the error control, status, and reporting flow for the PCIe devices/functions of the SoC.

### Table 6-10. Local Error Registers — SATA, USB, LPC and SMBus-Legacy

| Location in Host Root Configuration Space | SATA | USB Combo | LPC | SMBus-Legacy | Register Name |
|---|---|---|---|---|---|
| PCI Header Type 0 | CMD | CMD | CMD | CMD | Command |
| | STS | STS | STS | DS | Status |
| Device-Specific | SATAGC | XHCC1 | | | n/a |
| | | XHCC3 | | | n/a |
| | | USBSTS | | | n/a |

### Table 6-11. Local Error Registers — SPI and HSUART

| Location in Host Root Configuration Space | SPI | HSUART | Register Name |
|---|---|---|---|
| PCI Header Type 0 | BIOS_SPI_STS_CMD | CMD_STAT_REG | Command |
| | | | Status |
| Device-Specific | BIOS_SPI_UR_STS_CTL | DSC | n/a |

## 6.16 Error Reporting Flow — Non-PCI devices

These internal devices do not use the PCI or PCIe error-reporting mechanisms. These devices are part of the SoC UnCore (System Agent) and power control circuitry.

- System Agent - A Unit
- System Agent - B Unit
- System Agent - D Unit 0
- System Agent - D Unit 1
- SoC Power Management Unit (PMU)

The A, B, and two D Units of the SoC System Agent, also referred to as the "UnCore," protect the internal data interfaces with parity. Temporarily-stored data are protected with byte parity. The UnCore reports internal hardware errors by generating Machine Check Errors (MCERR) to the CPUs (MC Banks 6, 4, 7, and 8 respectively). The MCERR can also be routed to the MCERR_N output signal pin of the SoC. See Figure 6-5, "MCERR and IERR Handling." System Agent detected errors can also be routed directly to the SoC Global Error logic and logged in the SoC Global Error registers. See Figure 6-6, "Local and Global Error Register Structure."

The SoC Power Management Unit (PMU) detects Internal Errors (IERR) and can generate Machine Check events (MC Bank 5) to the CPUs. The IERR can also be routed to the IERR_N output signal pin of the SoC. See Figure 6-5, "MCERR and IERR Handling."

The D Units 0 and 1 detect and correct single-bit ECC (Correctable) errors. These can generate CMCI. See Section 6.6.2.3, "Corrected Machine-Check error Interrupt (CMCI)."

The SoC detects and logs Correctable Errors and Uncorrectable Errors encountered during DRAM memory Patrol Scrubbing and Demand Scrubbing. These can generate a Machine Check event and MCERR_N output signal. As an option, these errors can also be routed to the Uncorrectable (Double Bit) Read Error, bit 1 (UCE), and the Correctable (Single Bit) Read Error, bit 0 (SBE), of the DRAM Error Status (DERRSTS) register. The DERRSTS register is located in Host Root Memory Space at SBREG_BAR plus offset 10_0124h. For Memory Controller 1, the DERRSTS register is located at SBREG_BAR, offset 12_0124h. From there they can be routed to the SoC Global Error logic/registers and generate an NMI or SMI, and optionally assert the SoC ERROR_N[2:0] output signal pins.

PMU hardware errors also can generate Machine Check events and generate the IERR_N and MCERR_N signal pins. However, the PMU errors are not reported directly to the Global Error logic/registers. Instead they are reported by the Integrated Error Handler (IEH) through the GLREG (B0:D4:F0). This is similar to the flow used by the SoC I/O fabric. See Section 6.17, "Error Reporting Flow — I/O Fabric."

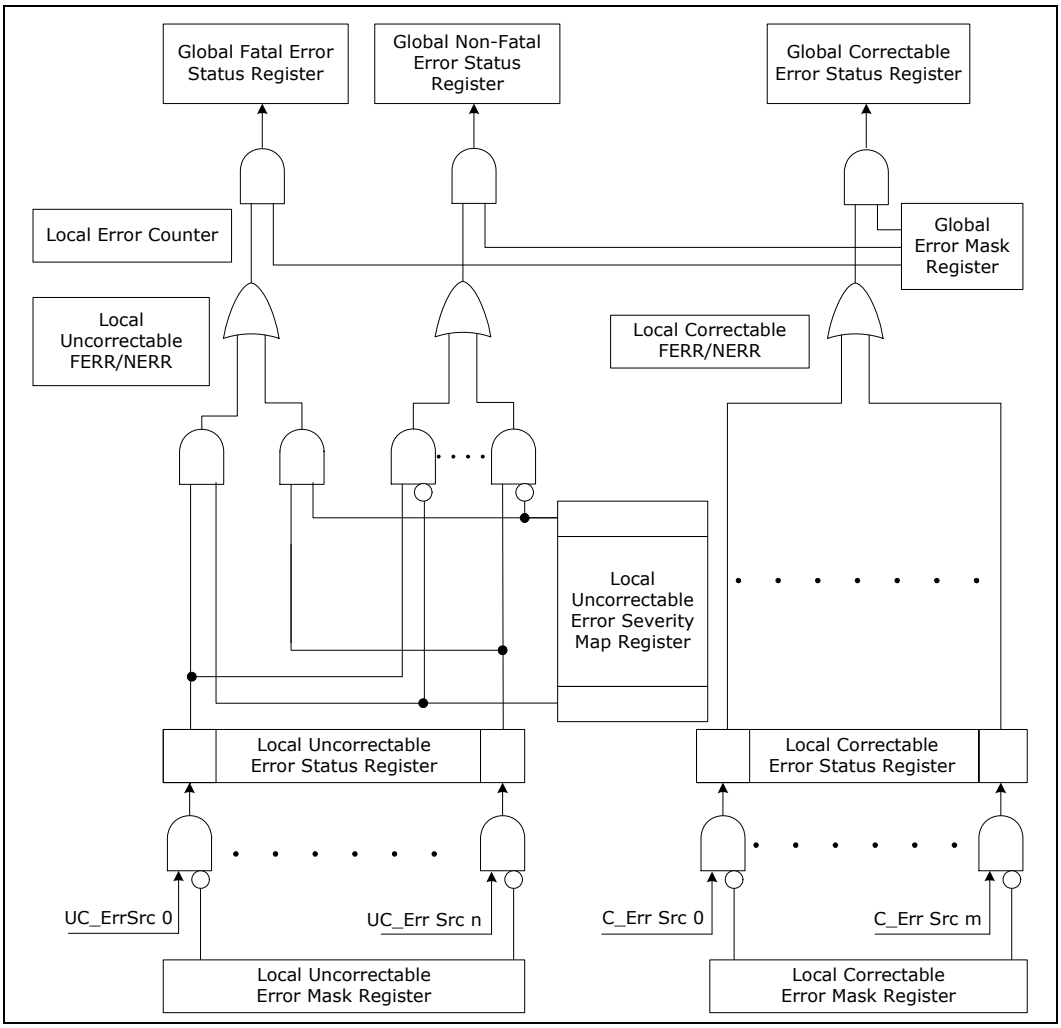
## 6.16.1 Local Error Registers

Both Correctable Errors and Uncorrectable Errors are reported and logged at the local level. See Figure 6-10, "Error Signaling and Logging for Non-PCI Devices."

The Uncorrectable Error Severity Register further categorizes the error as a Fatal Error or Non-Fatal Error. Error reporting can be masked. The First Error and Next Error are logged for each unit and a selectable error counter register is available.

The Local Error registers are listed in Table 6-12, "Local Error Registers — A-Unit and B-Unit" and Table 6-13, "Local Error Registers — D-Units."

**Figure 6-10. Error Signaling and Logging for Non-PCI Devices**



The local errors of the A-Unit, B-Unit, and the two D-Units are units are accumulated and directed to bits of the SoC Global Error Registers as shown in Figure 6-6, "Local and Global Error Register Structure."

The A-Unit and B-Unit Local Error registers and logs are located in Host Root Memory Space as offset addresses from MCHBAR:

- A-Unit: Offset 6588h through 65B8h
- B-Unit: Offset 6E04h through 6E3Ch

**Table 6-12. Local Error Registers — A-Unit and B-Unit**

| A-Unit | B-Unit | Register Name |
|---|---|---|
| A_CR_UNCERRSTS_0_0_0_MCHBAR | B_CR_UNCERRSTS_0_0_0_MCHBAR | Uncorrectable Error Status Register |
| A_CR_UNCERRMSK_0_0_0_MCHBAR | B_CR_UNCERRMSK_0_0_0_MCHBAR | Uncorrectable Error Mask Register |
| A_CR_UNCERRSEV_0_0_0_MCHBAR | B_CR_UNCERRSEV_0_0_0_MCHBAR | Uncorrectable Error Severity Register |
| A_CR_UNCFERRSTS_0_0_0_MCHBAR | B_CR_UNCFERRSTS_0_0_0_MCHBAR | First Uncorrectable Error Register |
| A_CR_UNCNERRSTS_0_0_0_MCHBAR | B_CR_UNCNERRSTS_0_0_0_MCHBAR | Next Uncorrectable Error Register |
| A_CR_UNCERRCNTSEL_0_0_0_MCHBAR | B_CR_UNCERRCNTSEL_0_0_0_MCHBAR | Uncorrectable Error Select Register |
| A_CR_UNCERRCNT_0_0_0_MCHBAR | B_CR_UNCERRCNT_0_0_0_MCHBAR | Uncorrectable Error Count |
| A_CR_CORERRSTS_0_0_0_MCHBAR | B_CR_CORERRSTS_0_0_0_MCHBAR | Correctable Error Status Register |
| A_CR_CORERRMSK_0_0_0_MCHBAR | B_CR_CORERRMSK_0_0_0_MCHBAR | Correctable Error Mask Register |
| A_CR_CORFERRSTS_0_0_0_MCHBAR | | First Correctable Error Register |
| A_CR_CORNERRSTS_0_0_0_MCHBAR | | Next Correctable Error Register |
| A_CR_CORERRCNTSEL_0_0_0_MCHBAR | | Correctable Error Count Select |
| A_CR_CORERRCNT_0_0_0_MCHBAR | | Correctable Error Count |
| | B_CR_DPHDRLOG_LO_0_0_0_MCHBAR | Data Parity Header Log |
| | B_CR_DPHDRLOG_HI_0_0_0_MCHBAR | |
| | B_CR_DPADDRLOG_LO_0_0_0_MCHBAR | Data Parity Address Log |
| | B_CR_DPADDRLOG_HI_0_0_0_MCHBAR | |
| | B_CR_DERRADDRLOG_LO_0_0_0_MCHBAR | Error Address Log |
| | B_CR_DERRADDRLOG_HI_0_0_0_MCHBAR | |
| A_CR_PARITY_CTL_0_0_0_MCHBAR | B_CR_PARITY_CTL_0_0_0_MCHBAR | Parity and Error Injection Control |
| A_CR_IOMMU_PARITYINJ | | Parity Error Injection |

The D-Units report single-bit errors and double-bit errors that occur during reads of system memory. SoC internal buffer errors and DDR interface command and address parity errors are also reported. Memory errors are reported when Demand Scrub and Patrol Scrub are performed. D-Unit0 is Memory Controller 0 and D-Unit1 is Memory Controller 1. The Local Error registers are listed in Table 6-13, "Local Error Registers — D-Units."

The D-Unit Local Error registers are located in Host Root Memory Space as offset addresses from SBREGBAR:

- D-Unit0: Offset 10_0120h through 10_0130h
- D-Unit1: Offset 12_0120h through 12_0130h

The D-Unit Local Error logs and counters are located in Host Root Memory Space as offset addresses from SBREGBAR:

- D-Unit0: Offset 10_0064h through 10_0090h
- D-Unit1: Offset 12_0064h through 12_0090h

### Table 6-13. Local Error Registers — D-Units

| D-Unit 0 | D-Unit 1 | Register Name |
|---|---|---|
| UCELOG | UCELOG | Uncorrectable Error Log |
| UCEADDR | UCEADDR | Uncorrectable Error Address |
| SBELOG | SBELOG | Correctable (Single Bit) Error Log |
| SBEADDR | SBEADDR | Correctable Error Address |
| SBECNT0 | SBECNT0 | SBE Count for Rank 0 |
| SBECNT1 | SBECNT1 | SBE Count for Rank 1 |
| SBECNT2 | SBECNT2 | SBE Count for Rank 2 |
| SBECNT3 | SBECNT3 | SBE Count for Rank 3 |
| SBEACC0 | SBEACC0 | SBE Accumulator for Rank 0 |
| SBEACC1 | SBEACC1 | SBE Accumulator for Rank 1 |
| SBEACC2 | SBEACC2 | SBE Accumulator for Rank 2 |
| SBEACC3 | SBEACC3 | SBE Accumulator for Rank 3 |
| FERRNERR | FERRNERR | First Error and Next Error Register |
| DERRSTS | DERRSTS | Error Status Register |
| DERRMSKSEV | DERRMSKSEV | Error Mask and Severity |
| DERRCNTSEL | DERRCNTSEL | Uncorrectable Error Count Select |
| DERRCNT | DERRCNT | Uncorrectable Error Count |
| ERRINJCTL | ERRINJCTL | Error Injection Control |
| ERRINJADDR | ERRINJADDR | Error Injection Address |
| ERRINJAMSK | ERRINJAMSK | Error Injection Address Mask |
| ERRINJDATA0 | ERRINJDATA0 | Error Injection Data 0 |
| ERRINJDATA1 | ERRINJDATA1 | Error Injection Data 1 |

### 6.16.2 Error Injection Control and Registers — Memory Controller

The SoC provide an error-injection mechanism for the D-Unit. The control, address, mask and data for this mechanism is located in Host Root Memory Space as offset addresses from SBREGBAR:

- D-Unit0: Offset 10_0134h through 10_0144h
- D-Unit1: Offset 12_0134h through 12_0144h

The error-injection registers are listed in Table 6-13, "Local Error Registers — D-Units."

## 6.17 Error Reporting Flow — I/O Fabric

The internal I/O fabric is parity protected. Both the command and data paths of the internal I/O fabric are end-to-end parity protected. The System Agent checks the I/O fabric parity on the command when a command is received from an I/O agent. If a command parity error is detected the packet is dropped and an error is reported. In addition, the System Agent does not accept new requests from the faulty I/O agent.

When an I/O agent receives a request from the System Agent, the I/O agent checks for both command and data parity. If a command parity error is detected, the agent drops the request and logs the error.

For Root Ports, if a data parity error is detected from a request from the System Agent, the Root Port forwards the error (also known as data poisoning) by setting the set the EP bit in the Transaction-Level Packet (TLP) of the request and forwards the request downstream over the PCI Express link. The Root Port also sets bit 15, Detected Parity Error (DPE), of its Primary Status Register (offset 6h) and sets bit 12, Poisoned TLP Status, of its Uncorrectable Error Status Register that is part of the AER capability structure. The Root Port indicates this non-fatal event to the Root Complex by sending a ERR_COR message with bit 13, Advisory Non-Fatal Error Status, of its AER Correctable Error Status Register.

### 6.17.1 Local Error Registers

The I/O fabric Local Errors are reported by the Integrated Error Handler (IEH) through the GLREG (B0:D4:F0) device-specific registers in Host Root Configuration Space starting at offset 280h. All I/O fabric errors are Uncorrectable Errors. See Table 6-14, "Local Error Registers — Integrated Error Handler (IEH)." All of these Uncorrectable Errors are also reported to the RCEC (B0:D5:F0) using Uncorrectable Internal Error (UIE) status (Bit 22) of the RCEC_ERRUNCSTS register.

**Table 6-14. Local Error Registers — Integrated Error Handler (IEH)**

| Location in Host Root Configuration Space | GLREG | Register Name |
|---|---|---|
| Device-Specific | GLREG_LERRUNCSTS | Uncorrectable Error Status |
| | GLREG_LERRUNCMSK | Uncorrectable Error Mask |
| | GLREG_LFERRUNCSTS | Uncorrectable First Error Status |
| | GLREG_LNERRUNCSTS | Uncorrectable Next Error Status |
| | GLREG_LUNCERRCNTSEL | Uncorrectable Error Counter Selection |
| | GLREG_LUNCERRCNT | Uncorrectable Error Counter |
| | GLREG_LFERRHDRLOG1 | Header Log (one entry) |
| | GLREG_LFERRHDRLOG2 | |
| | GLREG_LFERRHDRLOG3 | |
| | GLREG_LFERRHDRLOG4 | |

## 6.18 SoC Error Handling Summary

Table 6-15, "Summary of Default Error Logging and Responses" summarizes many of the SoC-detected hardware errors, their error severity, and the Local and Global Error registers used to report and log them.

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 1 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **Integrated Error Handler Errors** | | | | |
| A0 | IEH Detected Configuration Register Parity Error | Uncorrectable (Fatal) | The SoC detects and logs the error. | FERR/NERR is logged in IEH and Global Fatal Error Log registers:<br><br>GLREG_LERRUNCSTS<br>GLREG_LFERRUNCSTS<br>GLREG_LNERRUNCSTS<br>GLREG_LFERRHDRLOG<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME<br><br>Header is logged. |
| A1 | Configuration agent detected Fabric Command Parity Error | | Internal I/O Bus Command Parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A2 | Configuration Agent Fabric Data Parity Error | | SoC detects and logs the error. | |
| A3 | Configuration agent detected negatively decoded transaction Error | | SoC detects and logs the error. | |
| A4 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A5 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A6 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A7 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A8 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A9 | Internal Fabric detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |
| A10 | Internal Fabric Intel® ME detected Command Parity Error | | Internal fabric command parity errors are detected at the output of the upstream and downstream queues when a request is granted. Further arbiter grants are inhibited. | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 2 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **B-Unit Errors** | | | | |
| **B0** | Read data correctable ECC error | Correctable | Read data is corrected and written back to memory | FERR/NERR is logged in Bunit and Global Fatal Error Log Registers:<br><br>BCORERRSTS<br>BCORFERRSTS<br>BCORNERRSTS<br><br>GLREG_GCOERRSTS<br>GLREG_GCOFERRSTS<br>GLREG_GCONERRSTS<br>GLREG_GCOFERRTIME<br><br>These are also logged in Bunit MCA bank registers. |
| **B1** | Unexpected Completion | Uncorrectable (Fatal) | SoC detects and logs the error. | FERR/NERR is logged in Bunit and Global Fatal Error Log Registers:<br>BUNCERRSTS<br>BUNCFERRSTS<br>BUNCNERRSTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME<br><br>These are also logged in Bunit MCA bank registers. |
| **B2** | BRAM Write Data Parity Error | | | |
| **B3** | Memory Controller to SSA Data Parity Error | | | |
| **B4** | Source Address Decoder (SAD) Error | | | |
| **B5** | Ucode/xucode Errors | | | |
| **B6** | Read data uncorrectable ECC Error | | Read data is sent with bad parity to the B-Unit. | |
| **B8** | Protected Memory region space violated Error | | Protected memory violation | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 3 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **LAN Controllers Errors** | | | | |
| **C0** | Advisory Non-fatal Error (ANFE) | Correctable | | Error is logged in GbE Local and Global Correctable Error Log registers:<br><br>CES<br><br>GCORERRSTS<br>GCORFERRSTS<br>GCORNERRSTS<br>GCORFERRTIME |
| **C1** | Bad TLP Error (PTLPE) | Uncorrectable (Non-Fatal) | ERR_Non-fatal sent to root complex. Header is logged. A poisoned completion is ignored and the request can be retried after timeout. If enabled, the error is reported. | Error is logged in GbE Local and Global Non-fatal Error Log Registers:<br><br>UES<br><br>GNERRST<br>GNFERRST<br>GNNERRST<br>GNFERRTIME |
| **C2** | Completion Timeout Error (CTE) | | Error severity is non-fatal (default case): Send error message If advisory Retry the request once and send advisory error message on each failure. If fails, send uncorrectable error message. Error severity is defined as fatal: Send uncorrectable error message | |
| **C3** | Completer Abort Error (CAE) | | ERR_Non-fatal sent to root complex. Header is logged. Send completion with CA. | |
| **C4** | Unexpected Completion Error (UCE) | | ERR_Non-fatal sent to root complex. Header is logged. Discard TLP. | |
| **C5** | Unsupported Request Error (URE) | | ERR_Non-fatal sent to root complex. Header is logged. Send completion with UR. | |
| **C6** | Malformed TLP Error (MTLPE) | Uncorrectable (Fatal) | ERR_Fatal sent to root complex. Header is logged. Drop the packet and free FC credits. | Error is logged in GbE Local and Global Fatal Error Log Registers:<br>UES<br><br>GFERRST<br>GFFERRST<br>GFNERRST<br>GFFERRTIME |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 4 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **Memory Controller Errors** | | | | |
| **D0** | Patrol Scrub/Demand scrub correctable ECC Error | Correctable | Read data is corrected and written back to memory | FERR/NERR is logged in Dunit and Global Fatal Error Log Registers:<br><br>DERRSTS<br>DFERRNERR<br>SBELOG<br><br>UCELOG<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME<br><br>These are also logged in Dunit MCA bank registers. |
| **D1** | Write Data Buffer read parity Error | Uncorrectable (Fatal) | | |
| **D2** | Patrol scrub uncorrectable ECC Error | | Read data is sent with bad parity to the B-Unit. | |
| **D3** | Demand read uncorrectable ECC Error | | Read data is sent with bad parity to the B-Unit. | |
| **D4** | DDR4 Add/Cmd parity Error | | | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 5 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **PCIe* Root Port Errors** | | | | |
| E0 | Receiver Error (RE) | Correctable | Respond per *PCI-E Specification* | Log error per PCI Express* AER requirements for these correctable errors/message.<br><br>If the PCIe correctable error is forwarded to the Global Error registers, the error is logged in Global Correctable Log registers:<br>GCOERRST<br>GCOFERRST<br>GCONERRST<br>GLREG_GCOFERRTIME |
| E1 | Bad TLP Error (BTLPE) | | | |
| E2 | Bad DLLP Error (BDLLPE) | | | |
| E3 | Replay Number Rollover Error (RNRE) | | | |
| E4 | Replay Timer Time-out Error (RTTE) | | | |
| E5 | Header Log Overflow Error (HLOE) | | | |
| E6 | Received ERR_COR Message from Downstream Device | | | |
| E7 | Poisoned TLP Error (PTLPE) | Uncorrectable (Non-Fatal) | SoC logs the error. | Log error per PCI Express AER requirements for the corresponding error/message.<br><br>If the PCIe uncorrectable error is forwarded to the Global Error registers, the error is logged in Global Non-Fatal Log registers:<br><br>GNFERRST<br>GNFFERRST<br>GNFNERRST<br>GNFFERRTIME |
| E8 | Completion Time-out Error (CTE) | | | |
| E9 | Completer Abort Error (CAE) | | | |
| E10 | Unexpected Completion Error (UCE) | | Respond Per *PCI-E Specification* | |
| E11 | Unsupported Request Error (URE) | | SoC logs the error. | |
| E12 | ACS Violation Error (ACSE) | | | |
| E13 | MC Blocked TLP Error (MCE) | | | |
| E14 | Atomic Egress Blocked Error (AEBE) | | | |
| E15 | Received ERR_NONFATAL Message from Downstream Device | | Respond per *PCI-E Specification* | |
| E16 | Data Link Protocol Error (DLPE) | Uncorrectable (Fatal) | Respond per *PCI-E Specification* | Log error per PCI Express AER requirements for the corresponding error/message.<br><br>If the PCIe uncorrectable error is forwarded to the Global Error registers, the error is logged in Global Fatal Log registers:<br><br>GFAERRST<br>GFAFERRST<br>GFANERRST<br>GLREG_GFAFERRTIME |
| E17 | Surprise Link Down Error (SLDE) | | | |
| E18 | Flow Control Error (FCE) | | | |
| E19 | Receiver Overflow Error (ROE) | | | |
| E20 | Malformed TLP Error (MTLPE) | | | |
| E21 | Uncorrectable Internal Error (UIE) | | | |
| E22 | Received ERR_FATAL Message from Downstream Device | | | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 6 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **SMBus Errors** | | | | |
| **F0** | Retry Error (RETRYERR): An error due to SMT master transaction exceeding (non-collision) retry count as specified in RPOLICY.RETRY | Uncorrectable (Fatal) | The SoC detects and logs the error. | FERR/NERR is logged in the Global Fatal Error Log registers:<br><br>SMT_ERRUNCSTS<br>SMT_FERRUNCSTS<br>SMT_NERRUNCSTS<br>SMT_FERRHDRLOG<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 7 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **Intel® QAT Errors** | | | | |
| **G0** | Advisory Non-fatal Error (ANFE) | Correctable | ERR_COR sent to root complex | Error is logged in Intel® QAT Local and Global Correctable Error Log Registers:<br>PPAERCS/VPAERCS<br><br>GCORERRST<br>GCORFERRST<br>GCORNERRST<br>GCORFERRTIME |
| **G1** | Corrected Internal Error (CIE) | | | |
| **G2** | Poisoned TLP Error (PTLPE) | Uncorrectable (Non-fatal) | Poisoned completions: ERR_COR<br>Posted Write:<br>ERR_NONFATAL<br>NP Write: ERR_NONFATAL; Completion returned with UR | Error is logged in Intel® QAT Local and Global Non-fatal Error Log Registers:<br><br>PPAERUCS/VPAERUCS<br>PPAERHDRLOG/VPAERHDRLOG<br><br>GNERRST<br>GNFERRST<br>GNNERRST<br>GNFERRTIME |
| **G3** | Completion Timeout Error (CTE) | | ERR_COR sent to root complex | |
| **G4** | Completer Abort Error (CAE) | | P: request: ERR_NONFATAL NP request: ERR_COR; return completion with CA status | |
| **G5** | Unexpected Completion Error (UCE) | | ERR_COR; completion is dropped | |
| **G6** | Unsupported Request Error (URE) | | P: ERR_NONFATAL NP: ERR_COR; return completion with UR status | |
| **G7** | Malformed TLP Error (MTLPE) | Uncorrectable (Fatal) | ERR_NONFATAL (if sev = 0) ERR_FATAL (if sev = 1); When a malformed packet is detected, the packet is dropped and the error is logged. | Error is logged in Intel® QAT Local and Global Fatal Error Log Registers:<br><br>PPAERUCS/VPAERUCS<br>PPAERHDRLOG/VPAERHDRLOG<br><br>GFERRST<br>GFFERRST<br>GFNERRST<br>GFFERRTIME |
| **G8** | Uncorrectable Internal Error (UIE) | | ERR_NONFATAL (if sev = 0) ERR_FATAL (if sev = 1) | |
| **SATA Errors** | | | | |
| **H0** | When a set event occurs to configuration bits SATAGC.URD=1, during SATAGC.URRE=1 and CMD.SEE=1 for controller 1. | Uncorrectable (Fatal) | Treat as UR and initiate DO_SERR | Error is logged in PCISTS and Global Fatal Error Log registers:<br><br>PCISTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |
| **H1** | When a set event occurs to configuration bits STS.DPE=1 during CMD.PEE=1 and CMD.SEE=1 for controller 1. | | For MMR/CFG access, no update on destination & UR for NP req. Error logged and Do_SERR initiated.<br><br>For SATA data downstream, poison CRC but no error logging or reporting | |
| **H2** | When a set event occurs to configuration bits STS.STA=1 during CMD.SEE=1 for controller 1. | | Initiate DO_SERR. | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 8 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **USB Errors** | | | | |
| **J0** | Received downstream completion with unsupported request status | Uncorrectable (Fatal) | Set PCI space STS.RMA, and;<br>Set MMIO space USBSTS.HSE.<br>Note that USBSTS.HSE = 1, and USBCMD.HSEE = 1 will trigger Out-Of-Band error indication from Fresco IP, and SoC I/O fabric Gasket will set STS.SSE and send SERR if XHCC1.RMTASERR = 1 and CMD.SERR = 1.<br>Drop cycles (return credits) and signal unsuccessful completion. | Error is logged in PCISTS and Global Fatal Error Log registers:<br><br>PCISTS, USBSTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |
| **J1** | Received downstream completion with completer abort status | | Set PCI space STS.RTA, and;<br>Set MMIO space USBSTS.HSE.<br>Note that USBSTS.HSE = 1, and USBCMD.HSEE = 1 will trigger Out-Of-Band error indication from Fresco IP, and SoC I/O fabric Gasket will set STS.SSE and send SERR if XHCC1.RMTASERR = 1 and CMD.SERR = 1.<br><br>Drop cycles (return credits) and signal unsuccessful completion. | |
| **J2** | Received downstream memory cycles when PMCS.DS = D3. | | | |
| **J3** | Received downstream memory cycles when CMD.MSE = 0. | | | |
| **J4** | Received downstream memory cycles which is out of range due to recent memory space/limit change. | | | |
| **J5** | Received downstream memory cycles crossing QW aligned address boundary. | | | |
| **J6** | Received downstream request with error poisoned. | | | |
| **J7** | Received downstream completion with error poisoned. | | | |
| **J8** | Internal fabric Data parity error | | The SoC detects and logs the error. | |
| **J9** | Internal fabric command parity error | | The SoC detects and logs the error. | |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 9 of 10)

| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **LPC Errors** | | | | |
| **K0** | LPC Sync Error | Uncorrectable (Fatal) | SoC detects and logs the error. | Error is logged in PCISTS and Global Fatal Error Log registers:<br><br>PCISTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |
| **HSUART Errors** | | | | |
| **L0** | HSUART error | Uncorrectable (Fatal) | The SoC detects and logs the error. Do_Serr initiated. | Error is logged in PCISTS and Global Fatal Error Log Registers:<br><br>PCISTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |

## Table 6-15. Summary of Default Error Logging and Responses (Sheet 10 of 10)

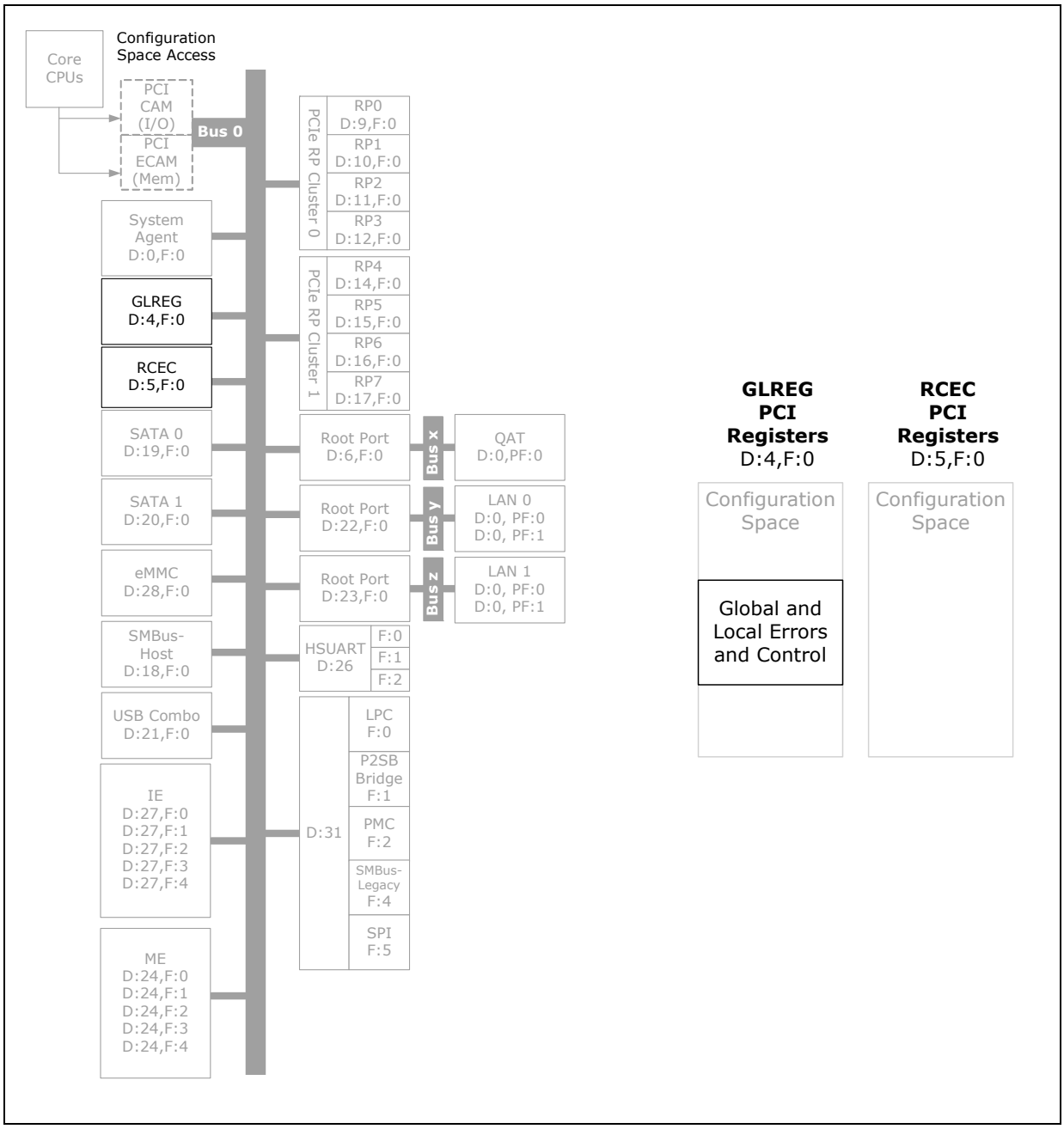
| ID | Error | Error Type (Default Severity) | Transaction Response | Default Error Logging[1] |
|---|---|---|---|---|
| **VT-d Errors** | | | | |
| I6 | Tag Array Parity Error during Context Cache Lookup | Correctable | A tag array parity error was detected in the VT-d caches. | Error is logged in Aunit Local and Global Correctable Error Log Registers:<br><br>GCORERRST<br>GCORFERRST<br>GCORNERRST<br>GCORFERRTIME |
| I7 | Tag Array Parity Error during L2 Lookup | | | |
| I8 | Tag Array Parity Error during L3 Lookup | | | |
| I9 | Tag Array Parity Error during L4 Lookup | | | |
| I10 | IOTLB Tag Array Parity Error | | | |
| I11 | Data Parity Error during Context Cache Lookup | Uncorrectable (Fatal) | A parity error was detected in the VT-d caches. | FERR/NERR is logged in Aunit and Global Fatal Error Log Registers:<br><br>AUNCERRSTS<br>AUNCFERRSTS<br>AUNCNERRSTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |
| I12 | Data Parity Error during L2 Lookup | | | |
| I13 | Data Parity Error during L3 Lookup | | | |
| I14 | Data Parity Error during L4 Lookup | | | |
| I15 | IOTLB Data Parity Error | | | |
| B8 | Protected Memory region space violated Error | Uncorrectable (Fatal) | Protected memory violation | FERR/NERR is logged in Bunit and Global Fatal Error Log Registers:<br><br>BUNCERRSTS<br>BUNCFERRSTS<br>BUNCNERRSTS<br><br>GLREG_GFAERRSTS<br>GLREG_GFAFERRSTS<br>GLREG_GFANERRSTS<br>GLREG_GFAFERRTIME |
| M0 | VT-d faults | Uncorrectable (Non-Fatal) | Log error per VT-d spec in Fault Record. | VTDBAR.FRCD |

1. This column notes the logging registers used assuming the error severity default remains. The error severity dictates the actual logging registers used upon detecting an error.

## 6.19 Register Map

Figure 6-11, "Register Map" shows the associated Global and Local Error Registers (GLREG) and the Root Complex Event Collector (RCEC) from a system software perspective.

**Figure 6-11. Register Map**



## 6.19.1 Registers in Configuration Space—GLREG

### Table 6-16. Registers in Host Configuration Space (Sheet 1 of 2)

| Configuration Space of B0:D4:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | GLREG_VID | GLREG Vendor ID |
| 2 | GLREG_DID | GLREG Device ID |
| 4 | GLREG_PCICMD | GLREG PCI Command |
| 6 | GLREG_PCISTS | GLREG PCI Status |
| 8 | GLREG_RID | GLREG Revision ID |
| 9 | GLREG_CC | GLREG Class Code |
| C | GLREG_CLS | GLREG Cacheline Size |
| 16 | GLREG_HDR | GLREG Header Type |
| 2C | GLREG_SVID | GLREG Subsystem Vendor ID |
| 2E | GLREG_SID | GLREG Subsystem ID |
| 34 | GLREG_CAPPTR | GLREG Capabilities Pointer |
| 3C | GLREG_INTL | GLREG Interrupt Line |
| 3D | GLREG_INTP | GLREG Interrupt Pin |
| 40 | GLREG_EXPCAPLST | GLREG PCI Express Capability List |
| 42 | GLREG_EXPCAP | GLREG PCI Express Capabilities |
| 44 | GLREG_DEVCAP | GLREG Device Capabilities |
| 48 | GLREG_DEVCTL | GLREG Device Control |
| 4A | GLREG_DEVSTS | GLREG Device Status |
| 200 | GLREG_GCOERRSTS | GLREG Global Correctable Error Status |
| 204 | GLREG_GNFERRSTS | GLREG Global Non-Fatal Error Status |
| 208 | GLREG_GFAERRSTS | GLREG Global Fatal Error Status |
| 20C | GLREG_GERRMSK | GLREG Global Error Mask |
| 210 | GLREG_GCOFERRSTS | GLREG Correctable FERR Status |
| 214 | GLREG_GCONERRSTS | GLREG Global Correctable NERR Status |
| 218 | GLREG_GNFFERRSTS | GLREG Global Non-Fatal FERR Status |
| 21C | GLREG_GNFNERRSTS | GLREG Global Nonfatal NERR Status |
| 220 | GLREG_GFAFERRSTS | GLREG Global Fatal FERR Status |
| 224 | GLREG_GFANERRSTS | GLREG Global Fatal NERR Status |
| 228 | GLREG_GTIME | GLREG Global Error Timer |
| 230 | GLREG_GCOFERRTIME | GLREG Global Correctable FERR Error Time Stamp |
| 238 | GLREG_GNFFERRTIME | GLREG Global Nonfatal FERR Error Time Stamp |
| 240 | GLREG_GFAFERRTIME | GLREG Global Fatal FERR Error Time Stamp |
| 248 | GLREG_GSYSEVTSTS | GLREG Global System Event Status |
| 24C | GLREG_GSYSEVTMSK | GLREG Global System Event Mask |
| 250 | GLREG_GSYSEVTMAP | GLREG Global System Event Map |
| 254 | GLREG_ERRPINCTRL | GLREG Error Pin Control |

## Table 6-16. Registers in Host Configuration Space (Sheet 2 of 2)

| Configuration Space of B0:D4:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 258 | GLREG_ERRPINSTS | GLREG Error Pin Status |
| 25C | GLREG_ERRPINDATA | GLREG Error Pin Data |
| 280 | GLREG_LERRUNCSTS | GLREG Uncorrectable Error Status |
| 284 | GLREG_LERRUNCMSK | GLREG Uncorrectable Error Mask |
| 288 | GLREG_LFERRUNCSTS | GLREG Uncorrectable First Error Status |
| 28C | GLREG_LNERRUNCSTS | GLREG Uncorrectable Next Error Status |
| 290 | GLREG_LUNCERRCNTSEL | GLREG Uncorrectable Error Counter Selection |
| 294 | GLREG_LUNCERRCNT | GLREG Uncorrectable Error Counter |
| 298 | GLREG_LFERRHDRLOG1 | GLREG Header Log |
| 29C | GLREG_LFERRHDRLOG2 | GLREG Header Log |
| 2A0 | GLREG_LFERRHDRLOG3 | GLREG Header Log |
| 2A4 | GLREG_LFERRHDRLOG4 | GLREG Header Log |

## 6.19.2 Registers in Configuration Space—RCEC

### Table 6-17. Registers in Host Configuration Space (Sheet 1 of 2)

| Configuration Space of B0:D5:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | RCEC_VID | RCEC Vendor ID |
| 2 | RCEC_DID | RCEC Device ID |
| 4 | RCEC_PCICMD | RCEC PCI Command |
| 6 | RCEC_PCISTS | RCEC PCI Status |
| 8 | RCEC_RID | RCEC Revision ID |
| 9 | RCEC_CC | RCEC Class Code |
| C | RCEC_CLS | RCEC Cacheline Size |
| E | RCEC_HDR | RCEC Header Type |
| 2C | RCEC_SVID | RCEC Subsystem Vendor ID |
| 2E | RCEC_SID | RCEC Subsystem ID |
| 34 | RCEC_CAPPTR | RCEC Capabilities Pointer |
| 3C | RCEC_INTL | RCEC Interrupt Line |
| 3D | RCEC_INTP | RCEC Interrupt Pin |
| 40 | RCEC_EXPCAPLST | RCEC PCI Express Capability List |
| 42 | RCEC_EXPCAP | RCEC PCI Express Capabilities |
| 44 | RCEC_DEVCAP | RCEC Device Capabilities |
| 48 | RCEC_DEVCTL | RCEC Device Control |
| 4A | RCEC_DEVSTS | RCEC Device Status |
| 5C | RCEC_ROOTCTL | RCEC Root Control |
| 5E | RCEC_ROOTCAP | RCEC Root Capabilities |
| 60 | RCEC_ROOTSTS | RCEC Root Status |
| 80 | RCEC_PMCAPLST | RCEC Power Management Capability List |
| 82 | RCEC_PMCAP | RCEC Power Management Capabilities |
| 84 | RCEC_PMCSR | RCEC Power Management Control / Status |
| 90 | RCEC_MSICAPLST | RCEC MSI Capability List |
| 92 | RCEC_MSICTL | RCEC MSI Message Control |
| 94 | RCEC_MSIADDR | RCEC MSI Message Address |
| 98 | RCEC_MSIDATA | RCEC MSI Message Data |
| 9C | RCEC_MSIMSK | RCEC MSI Mask Bit |
| A0 | RCEC_MSIPENDING | RCEC MSI Pending Bit |
| 100 | RCEC_AERCAPHDR | RCEC Advanced Error Reporting Extended Capability Header |
| 104 | RCEC_ERRUNCSTS | RCEC Uncorrectable Error Status |
| 108 | RCEC_ERRUNCMSK | RCEC Uncorrectable Error Mask |
| 10C | RCEC_ERRUNCSEV | RCEC Uncorrectable Error Severity |
| 110 | RCEC_ERRCORSTS | RCEC Correctable Error Status |

**Table 6-17. Registers in Host Configuration Space (Sheet 2 of 2)**

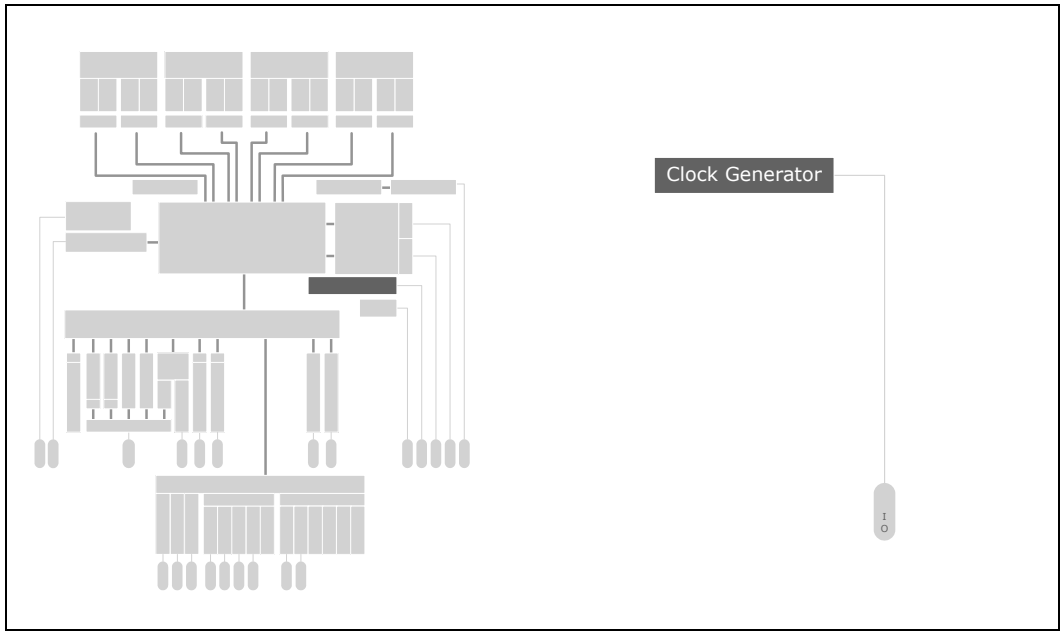
| Configuration Space of B0:D5:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 114 | RCEC_ERRCORMSK | RCEC Correctable Error Mask |
| 118 | RCEC_AERCAPCTL | RCEC Advanced Error Capabilities and Control |
| 11C | RCEC_AERHDRLOG1 | RCEC Header Log |
| 120 | RCEC_AERHDRLOG2 | RCEC Header Log |
| 124 | RCEC_AERHDRLOG3 | RCEC Header Log |
| 128 | RCEC_AERHDRLOG4 | RCEC Header Log |
| 12C | RCEC_ROOTERRCMD | RCEC Root Error Command |
| 130 | RCEC_ROOTERRSTS | RCEC Root Error Status |
| 134 | RCEC_ERRSRCID | RCEC Error Source Identification |
| 150 | RCEC_RCECEPACAPHDR | RCEC Root Complex Event Collector Endpoint Association Extended Capability Header |
| 154 | RCEC_ABMRCIEP | RCEC Association Bitmap for Root Complex Integrated Endpoints |

§ §

# 7 Clock Architecture

The SoC requires one external 25-MHz crystal and one external 32.768-kHz crystal for internal clock generation. Some internal clocks can be configured by Soft Straps and by registers accessible to UEFI IA Firmware (BIOS).

**Figure 7-1. What is Covered in This Chapter**



## 7.1 Signal Descriptions

The signal descriptions are shown in Table 7-1. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction/Type column of Table 7-1 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.
- Differential = Differential signal pair. Two pins: Positive; Negative.

**Table 7-1. Signal Names and Descriptions (Sheet 1 of 2)**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| CLK_X1_PAD | I | No | **Clock Crystal - 25 MHz**. The SoC uses the crystal for its internal oscillators and PLLs to generate internal clocks of various speeds which are controlled and distributed throughout the SoC. |
| CLK_X2_PAD | O | No | |
| ICLKRCOMP | I,O | No | **ISCLK RCOMP:** Compensation resistor for the CLK_OUT_DP[4:0], CLK_OUT_DN[4:0] differential output signals. The resistor is used by the internal ISCLK circuitry to calibrate the PCIe reference-clock driver impedance to comply with the PCIe specification for REFCLK. To achieve 50-Ω driver/trace impedance, ICLKRCOMP must be connected to a 200-Ω resistor to VSS on the platform board. To achieve a 42.5- Ω driver impedance/trace, a 169-Ω resistor must be used. |
| RTC_X1_PAD<br>RTC_X2_PAD | I,O | No | **Clock Crystal - 32.768 KHz**: The SoC uses the crystal for its internal Real-Time Clock (RTC) oscillator. The clock is used internally and drives the PMU_SUSCLK output signal. |
| BVCCRTC_EXTPAD | I,O | No | **External Pad for RTC Power Well:** Must be connected to an external 0.1-µF capacitor on the platform board. |
| FLEX_CLK_SE[1:0] | O | Yes | **Flex Clock.** These single-ended output signals are general-purpose clocks that can be used by the platform board. |
| PMU_SUSCLK | O | Yes | Chapter 16, "Power Management Controller (PMC)" |
| SVID_CLK | O-OD | Yes | Chapter 34, "Power Management" |
| SATA0_SCLK<br>SATA1_SCLK | O-OD | Yes | Chapter 12, "SATA Controller" |
| EMMC_CLK | I,O | Yes | Chapter 17, "Embedded Multi-Media Card (eMMC)" |
| DDR0_CLK_DP[3:0]<br>DDR0_CLK_DN[3:0]<br><br>DDR1_CLK_DP[3:0]<br>DDR1_CLK_DN[3:0] | O, Differential | No | Chapter 9, "Memory Controller" |
| CLK_OUT_DP[4:0]<br>CLK_OUT_DN[4:0] | O, Differential | No | Chapter 11, "PCI Express Root Ports (RP)" |
| LPC_CLKOUT[1:0] | O | Yes | Chapter 20, "Low Pin Count (LPC) Controller" |
| SMB_LAN_CLK | I,O-OD | Yes | Chapter 13, "LAN Controllers" |
| LAN0_PORT0_I2C_CLK<br>LAN0_PORT1_I2C_CLK<br>LAN1_PORT0_I2C_CLK<br>LAN1_PORT1_I2C_CLK | I,O-OD | Yes | Chapter 13, "LAN Controllers" |
| LAN_MDC | O | Yes | Chapter 13, "LAN Controllers" |
| NCSI_CLK_IN | I | Yes | Chapter 13, "LAN Controllers" |
| SPI_CLK | I,O | Yes | Chapter 19, "Serial Peripheral Interface (SPI)" |
| SMB_LEG_CLK | I,O-OD | Yes | Chapter 28, "SMBus Controller - Legacy" |
| SMB_HOST_CLK | I,O-OD | Yes | Chapter 29, "SMBus Controller - Host" |

## Table 7-1. Signal Names and Descriptions (Sheet 2 of 2)

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| SMB_PECI_CLK | I,O-OD | Yes | Chapter 30, "SMBus Controller - Platform Environment Control Interface" |
| SMB_ME_SMT0_CLK | I,O-OD | Yes | Chapter 36, "System Management" |
| SMB_ME_SMT1_CLK | | Yes | |
| SMB_ME_SMT2_CLK | | Yes | |
| SMB_IE_SMT0_CLK | I,O-OD | Yes | Chapter 37, "Innovation Engine" |
| SMB_IE_SMT1_CLK | | Yes | |
| SMB_IE_SMT2_CLK | | Yes | |
| TCK | I | No | Chapter 38, "JTAG and Debug Ports" |
| DFX_PORT_CLK[1:0] | O | Yes | Chapter 38, "JTAG and Debug Ports" |

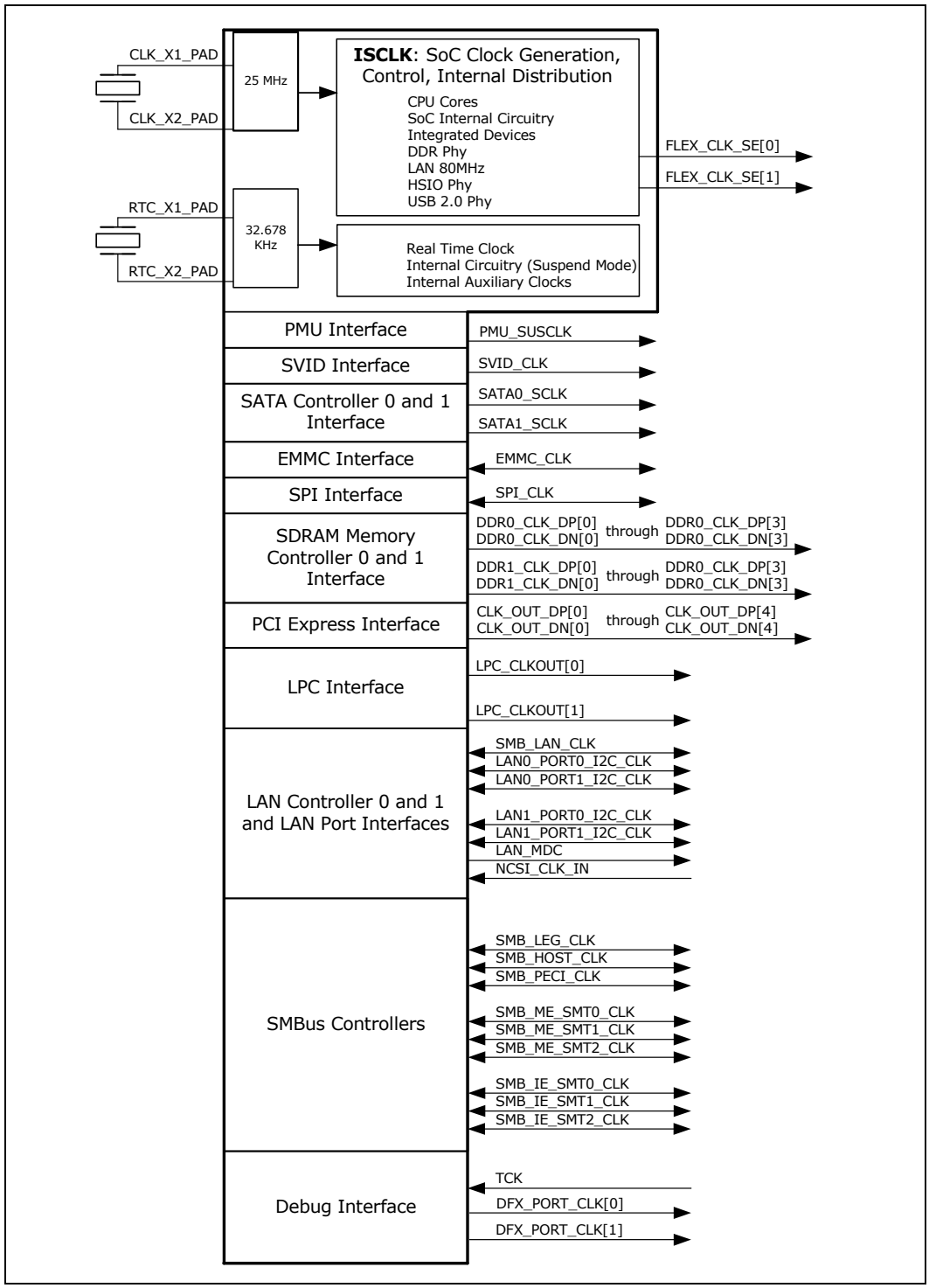
## 7.2 Feature List

- SoC requires only two crystals to generate the necessary reference clocks.
- Two Flex Clock output signals that are configurable for platform board use.
- Spread-Spectrum Clocking (SSC) and Non-Spread-Spectrum clocking (NSSC) available.
- Five PCI Express output reference clocks provided and can be used without additional buffer circuits.

## 7.3 Architectural Overview

The SoC clock architecture is shown in Figure 7-2.

**Figure 7-2. SoC Clock Architecture**



### 7.3.1 ISCLK

The external 25-MHz crystal serves as the reference for the internal block of clocks called the Internal System Clock (ISCLK). The reference is routed to a number of internal Phase-Locked Loop (PLL) circuits which are then distributed to the CPU cores, internal circuitry and devices.

For external-interface clocks generated by the SoC, both Spread-Spectrum Clocking (SSC) and Non-SSC (NSSC) are supported.

There are two customer-configurable clock features:

- **Spread Spectrum Controls:** This is expected to be enabled by UEFI IA Firmware via an Integrated Clock Control HECI command to Intel® ME.
- **Flex I/O Clock Speed:** The setting is programmable via a soft strap that is applied during reset. Software can read the resultant settings of the two Flex Clocks' speeds by reading bits [5:0] and [13:8] of the ICLK_DWORD8 register in Host Root Memory Space.

### 7.3.2 Real-Time Clock

The external 32.768-kHz crystal serves as the reference for the integrated Real Time Clock (RTC). It also provides the clock used by circuitry and devices when the SoC is in Suspend Mode (SUS Mode).

*Note:* All platform board designs must provide this crystal even if the RTC is not needed or SUS Mode is not implemented.

### 7.3.3 PMU SUS Clock

The SoC provides the PMU_SUSCLK output signal pin that is based on the 32.768-KHz RTC clock. This signal can be used by the platform board.

### 7.3.4 Flex Clocks

FLEX_CLK_SE[1:0] are two, single-ended, output signals available to the SoC customer as general-purpose clocks that can be used by the platform board. The customer uses Soft Straps (Chapter 4, "Strapping and Configuration") to enable and select one of the supported frequencies for each Flex Clock. The supported frequencies for each Flex Clock are:

- 25 MHZ
- 33 MHz
- 48 MHz
- 50 MHz

### 7.3.5 SVID Clock

The available clock frequencies for the SVID Clock:

- 16.7 MHZ
- 20 MHZ
- 25 MHz (default)

### 7.3.6 SATA Serial GPIO (SGPIO) Clock

The clock frequency for the SATA Serial Clocks is 32 KHz.

### 7.3.7 EMMC Clock

The SoC EMMC clock is 200 MHz.

### 7.3.8 DDR Clocks

The available clock frequencies for the DDR4 differential clock output signals are product-SKU dependent. The entire list follows:

- 933 MHz (1866 MHz SDRAMs)
- 1067 MHz (2133 MHz SDRAMs)
- 1200 MHz (2400 MHz SDRAMs)

### 7.3.9 PCI Express Reference Clocks

The clock frequency for the PCI Express* reference clocks is 100 MHz.

### 7.3.10 SPI Clock

The available clock frequencies for the SPI Clock:

- 17 MHz (default)
- 30 MHz
- 48 MHz

### 7.3.11 LPC Clocks

The SoC LPC clock is 24 MHz.

### 7.3.12 LAN I2C Clocks

The available clock frequencies for each of the LAN I2C Clocks are:

- 100 kHz
- 400 kHz
- 1 MHz

### 7.3.13 LAN MDC Clock

The available clock frequencies for the LAN MDIO management interface clock are:

- 240 kHz  (available only with 1 Gb/s Link Speed)
- 2.4 MHz

### 7.3.14 LAN NC-SI Clock

NCSI_CLK_IN is designed to be driven by the 50-MHz NC-SI generated by the platform board.

### 7.3.15 SMBus Clocks

The available clock frequencies for each of the SMBus controllers of the SoC are:

- 100 kHz
- 400 kHz
- 1 MHz

### 7.3.16 Debug and DFx Clocks

- The SoC TCK clock input signal can be 100 MHz maximum
- The SoC DFX_PORT_CLK[1:0] clocks are 667 MHz.

### 7.3.17 Clocks Generated During the S5 Sleep State

The internal clocks for the USB 3.0 PHY (integrated in the HSIO circuitry), USB Combo Controller, and LAN Controllers are generated even during the S5 Sleep State.

§ §

# 8 Interrupts and Events

Global interrupt handling is described in this chapter. The types of interrupts are described as well as the routing and mapping of these interrupts to the integrated I/O Advanced Programmable Interrupt Controller (APIC) and 8259 Programmable Interrupt Controller (PIC).

## 8.1 PCI Interrupts and Routing

The SoC does not provide external INTA# through INTD# PCI interrupt signal pins. The PCI Express* integrated endpoints and the PCIe* interrupt messages internally produce the equivalent of these four interrupt signals for the device. Since the SoC Interrupt and Timer Sub-System (ITSS) implements only 13 registers, some PCI Interrupt Route (PIR) registers must be shared by multiple devices.

To support sharing a PIR register among multiple interrupt sources, the ITSS tracks the interrupt assertion and deassertion separately for each source. An interrupt asserts with the first assertion message and remains asserted until the last deassertion message.

PCI Device numbers 0-4, 7-8, 13, and 28-29 are currently unused by the SoC. PCI Device 0, Device 4, Device 25 and Devices 28-29 do not issue INT[A-D] interrupt message. See following Table 8-1 for details.

**Table 8-1. PIRQ Routing Control Information For INT[A-D] Messages (Sheet 1 of 2)**

| PCI Device | PCI Interrupt Route Register: PIR[12:0] | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Device 0-4 | does not send legacy interrupts | | | | | | | | | | | | |
| Device 5<br>Fn0: RCEC | | | | **3** | | | | | | | | | |
| Device 6<br>Fn0: VRP2<br>(for Intel® QAT) | | | | | **4** | | | | | | | | |
| Device 7<br>Device 8 | are unused | | | | | | | | | | | | |
| Device 9<br>Fn0: PCIe RP[0] | | | | | | | | **7** | | | | | |
| Device 10<br>Fn0: PCIe RP[1] | | | | | | | | **7** | | | | | |
| Device 11<br>Fn0: PCIe RP[2] | | | | | | | | **7** | | | | | |
| Device 12<br>Fn0: PCIe RP[3] | | | | | | | | **7** | | | | | |
| Device 13 | is unused | | | | | | | | | | | | |
| Device 14<br>Fn0: PCIe RP[4] | | | | | | | | | **8** | | | | |
| Device 15<br>Fn0: PCIe RP[5] | | | | | | | | | **8** | | | | |
| Device 16<br>Fn0: PCIe RP[6] | | | | | | | | | **8** | | | | |
| Device 17<br>Fn0: PCIe RP[7] | | | | | | | | | **8** | | | | |

## Table 8-1. PIRQ Routing Control Information For INT[A-D] Messages (Sheet 2 of 2)

| PCI Device | PCI Interrupt Route Register: PIR[12:0] | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Device 18<br>Fn0: SMBus - Host | | | | | | | | | | | **10** | | |
| Device 19<br>Fn0: SATA0 | | | | | | | **6** | | | | | | |
| Device 20<br>Fn0: SATA1 | | | | | | | | | | | | **11** | |
| Device 21<br>Fn0: USB Combo | | | | | | | | | | **9** | | | |
| Device 22<br>Fn0: VRP0<br>(for LAN Ctrl0) | | **1** | | | | | | | | | | | |
| Device 23<br>Fn0: VRP1<br>(for LAN Ctrl1) | | | **2** | | | | | | | | | | |
| Device 24<br>Fn0: ME.HECI1<br>Fn1: ME.HECI2<br>Fn2: ME.PTIO-IDER<br>Fn3: ME.PTIO-KT<br>Fn4: ME.HECI3 | | | | | | **5** | | | | | | | |
| Device 25 | does not send legacy interrupts | | | | | | | | | | | | |
| Device 26<br>Fn0: HSUART0<br>Fn1: HSUART1<br>Fn2: HSUART2 | | | | | | | | | | | **10** | | |
| Device 27<br>Fn0: IE.HECI1<br>Fn1: IE.HECI2<br>Fn2: IE.PTIO-IDER<br>Fn3: IE.PTIO-KT<br>Fn4: IE.HECI3 | | | | | | | | | | | | | **12** |
| Device 28<br>Fn0: eMMC | | | | | | | | | | | **10** | | |
| Device 29 | are unused | | | | | | | | | | | | |
| Device 30 | | | | | | | | | | | | | |
| Device 31<br>Fn0: LPC<br>Fn1: P2SB<br>Fn4: SMBus - Legacy<br>Fn7: Intel® Trace Hub | **0** | | | | | | | | | | | | |

### 8.1.1 Interrupt Pin Route Overview

There are 13, 16-bit PCI interrupt routing registers. These 13 registers, named PCI Interrupt Route (PIR) 0-12, are located in memory space and sideband port PCI Interrupt Route 0 (PIR0)—Offset 3140h through PCI Interrupt Route 12 (PIR12)—Offset 3158h. Each 16-bit register has four 3-bit fields:

- Interrupt A Pin Route (IAR): INTA# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- Interrupt B Pin Route (IBR): INTB# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- Interrupt C Pin Route (ICR): INTC# mapping to one of PIRQA# through PIRQH# (0h through 7h)
- Interrupt D Pin Route (IDR): INTD# mapping to one of PIRQA# through PIRQH# (0h through 7h)

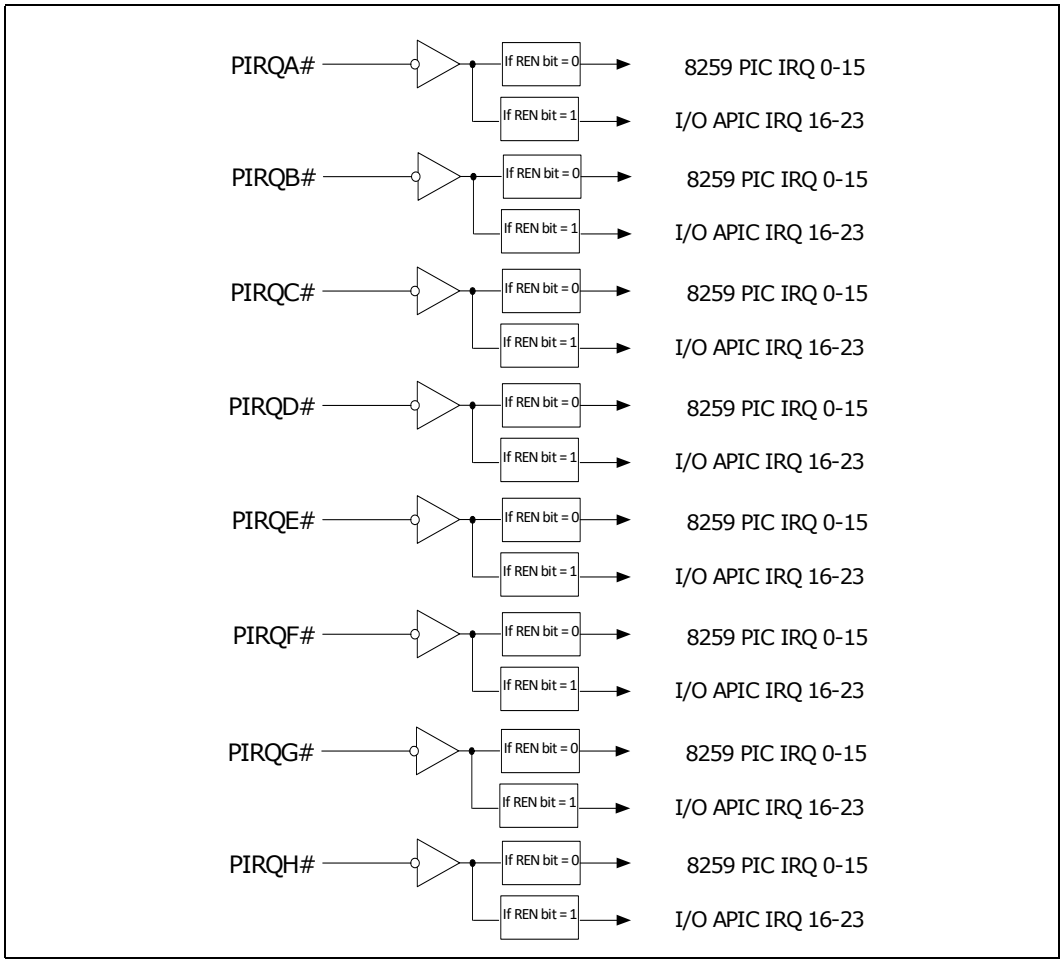
The field values 8h through Fh are invalid.

*Note:* Each of the eight PIRQx# can represent the interrupt of more than one device and up to four for each device.

PIRQA# through PIRQH# are enabled for routing to 8259 and steered to IRQ0-15; similar steering is done for IOxAPIC to IRQ16-23. Information about programming the 8259 Program Interrupt Controller (PIC) and I/O APIC Controller in Chapter 21, "PIC and I/O APIC Controllers".

The PIRQx is routed to the 8259 PIC mode provided that PIRQx Routing Control Register (PxRC) bit 7 Interrupt Routing Enable (REN) is programmed to 0. When REN is programmed as 1, the particular PIRQx is routed to the APIC mode.

**Figure 8-1. PIRQ Routing Control Overview**




*Note:* PIRQx#s are active-low interrupt sources. They are internally inverted before being sent to the PIC. Therefore, the interrupt level reference to the signals at the internal interface of the 8259's, the term "high" indicates "active".

## 8.1.2 PIRQx Register Decode Table

There are eight, 8-bit routing control registers called PIRQx Routing Control (PxRC), one for each PIRQA through PIRQH. Each of the eight interrupts are routed to one of 11 IRQ inputs of the integrated 8259 PIC according to PxRC bit 3:0 IRQ Routing (IR) field.

PIRQx Routing Control Register (PxRC) bit 7 Interrupt Routing Enable (REN) must be set to 0, the corresponding PIRQ is routed to the legacy 8259 mode. Otherwise, the PIRQ is routed to I/OxAPIC mode.

**Table 8-2. PIRQA through PIRQH Routing Register IRQ Decode**

| PIRQx Register, IR field 3:0 | PIRQx Routing to 8259 PIC |
|---|---|
| 0000 | Reserved |
| 0001 | Reserved |
| 0010 | Reserved |
| 0011 | IRQ3 of the 8259 PIC |
| 0100 | IRQ4 of the 8259 PIC |
| 0101 | IRQ5 of the 8259 PIC |
| 0110 | IRQ6 of the 8259 PIC |
| 0111 | IRQ7 of the 8259 PIC |
| 1000 | Reserved |
| 1001 | IRQ9 of the 8259 PIC |
| 1010 | IRQ10 of the 8259 PIC |
| 1011 | IRQ11 of the 8259 PIC |
| 1100 | IRQ12 of the 8259 PIC |
| 1101 | Reserved |
| 1110 | IRQ14 of the 8259 PIC |
| 1111 | IRQ15 of the 8259 PIC |

## 8.1.3 Edge- and Level-Triggered Mode

There are two edge/level triggered Control Register ELCR1 (Master) and ELCR2 (Slave) to control edge and level triggered mode selection per interrupt input.

If an ELCR bit is “0”, an interrupt request will be recognized by a low to high transition on the corresponding IRQ input. The IRQ input can remain high without generating another interrupt. If an ELCR bit is “1”, an interrupt request will be recognized by a high level on the corresponding IRQ input and there is no need for an edge detection. The interrupt request must be removed before the End of Interrupt (EOI) command is issued to prevent a second interrupt from occurring.

In both the edge and level triggered modes, the IRQ interrupt (called a spurious interrupt) must remain active until after the falling edge of the first internal INTA#. If the IRQ input goes inactive before this time, a default IRQ7 vector will be returned.

PIRQA# through PIRQH# are defined as level-sensitive interrupts. When routed to a specified IRQ line, the software must change the corresponding ELCR1 or ELCR2 register of the 8259 to a level-sensitive mode.

## 8.2 Non-Maskable Interrupt (NMI)

NMI is generated internally by serious system events, memory parity errors, some System Errors (SERR), and PCI Express fatal errors.

The NMI is reset by the software by setting the corresponding NMI source enable/disable bit in the NMI Status and Control (NMI_STS_CNT) register.

The 8-bit NMI_STS_CNT register is located in the I/O space at 61h. For the individual GPIO NMI status, enable and routing bit definitions, please see Chapter 26, "Customer General-Purpose I/O (GPIO)".

The NMI is broadcast to all cores in the SoC. Sleeping cores are first awakened. NMI Enable (NMI_EN) register is located in the I/O space at 70h.

The SoC provides interrupt mapping to generate the NMI for various SoC and system events. There is an NMI Control Register (NMI) to indicate current NMI status if any source is asserted. The register NMI is located as a Sideband register accessible in Host Memory Space at offset D0_3330h from the SBREG_BAR base address.

*Note:* ITSS may escalate other interrupts to NMI. ITSS also may demote NMI to SMI.

## 8.3 System Management Interrupt (SMI)

The SMI indicates any of several system-level conditions. Examples are:

- Thermal-sensor events
- Throttling activation
- System management RAM access
- Chassis open
- System power button pressed
- System Management Bus (SMBus) events
- Power Management Events (PME)
- PCI Express Hot-Plug* events
- Real Time Clock (RTC) alarm activation
- Various system-state-related activities

An SMI causes the system to enter System Management Mode (SMM). SMM is an operating mode in which all normal execution, including the operating system, is suspended, and special, separate software, usually firmware or a hardware-assisted debugger, is executed in high-privilege mode. The SMI is broadcast to all cores in the SoC. Sleeping cores are first awakened.

SMI Enable register is located in I/O space at 0x30h to enable SMI from different configurable source; SMI Status register is located in I/O space at 0x34h to show SMI status from different configurable source. For the individual GPIO SMI status, enable and routing bit definitions, please see Chapter 26, "Customer General-Purpose I/O (GPIO)".

## 8.4 System Control Interrupt (SCI)

SCI is a special type of hardware power-management interrupt that is handled directly by the OS, and is not handled by a device driver. It is closely tied to the ACPI model. The operating system uses the SCI interrupt to process ACPI events signaled by GPEs, whether the system is asleep or awake when the event occurs. If the system is in an S0 state, SCI can be caused; if the system is in the S5 state, SCI event will also wake the system.

If not using the I/O APIC for the SCI, the SCI must be routed to IRQ9-IRQ11 of the 8259 PIC. When routed to the 8259 PIC, the SCI is not sharable with the Serial Interrupt (SERIRQ) for that PIC input, but it is shareable with the PIRQA through PIRQH interrupts.

If using an I/O APIC, the SCI is mapped to the I/O APIC interrupt or to an SMI. Mapping SCI events to SMI can be used when a legacy OS is in use. The SCI Enable (SCI_EN) bit of the Power Management 1 Control (PM1_CNT) register controls whether the event is routed as an SCI or an SMI. The PM1_CNT register is located in the I/O space at PCI_Config_ABASE offset 4h.

The SCI routing to the I/O APIC is controlled by the 3-bit SCI IRQ Select (SCIS) field of the ACPI Control (ACTL) register located in the memory space at Bus 0, Device 31, Function 2 offset address 0x44h.

Also, if using the I/O APIC for SCI, the SCI can be mapped to IRQ20-IRQ23 of the I/O APIC, and can be shared with other interrupts. Here the SCI must be programmed for active-low reception. The SCI can also be mapped to IRQ9, IRQ10, of IRQ11 where it must be programmed for active-high reception.

**Table 8-3. Routing of SCI to the I/O APIC**

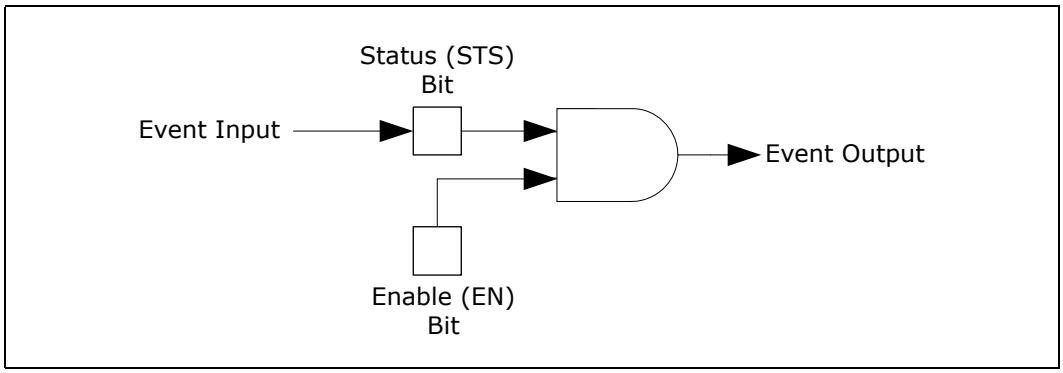
| ACTL.SCIS | SCI IRQ Select in 8259 | SCI IRQ Select in I/O APIC |
|---|---|---|
| 000 | IRQ9 | IRQ9 |
| 001 | IRQ10 | IRQ10 |
| 010 | IRQ11 | IRQ11 |
| 011 | | Reserved |
| 100 | | IRQ20 |
| 101 | | IRQ21 |
| 110 | | IRQ22 |
| 111 | | IRQ23 |

*Note:* Power Management Controller (PMC) may escalate SCI to SMI; otherwise SCI is delivered to ITSS as IRQx for handling.

## 8.5 GPIO and PMC General Purpose Event (GPE)

Some of the Customer GPIOs can be configured as SoC input-pin signals that can generate an ACPI General Purpose Event (GPE). The SoC Power Management Controller (PMC) can also generate ACPI GPE status. Each SoC GPE status can be enabled to generate a System Control Interrupt (SCI) or a System Management Interrupt (SMI). The GPE register model is shown in Figure 8-2. Notice that the Enable bit does not prevent the Status bit from setting when the event occurs.

**Figure 8-2. General Purpose Event (GPE) Register Model**




The GPIO controller has seven Dwords of GPE status and enable bits of which two Dwords originate from the GPIO North Community and five Dwords from the GPIO South Community.

Even though there are seven Dwords of GPIO GPE Status and seven Dwords of GPIO GPE Enable groups shown in Table 8-4, only three groups are exposed by the PMC. So BIOS must choose which three groups from the seven to use and make only these three GPE groups visible to software. These three GPIO groups are mapped to 96 sets of status/enable GPE0 register bits grouped as GPE0_DW0, GPE0_DW1 and GPE0_DW2.

The GPIO GPE Status and Enable groups are marked with group number in silicon. GPIO Miscellaneous Configuration Register (MISCCFG) located at

SBREG_BAR offset 0xC50010h and 0xC20010h bit 8-19 shows the default value is group 0, 1, and 2 mapping to GPE0_DW0, GPE0_DW1 and GPE0_DW2. By programming MISCCFG bit 8-19, different group can be assigned to GPE0_DW0, GPE0_DW1 and GPE0_DW2.

**Table 8-4. GPE0_DW Group Mapping**

| | | | |
|---|---|---|---|
| 0 | GPI_GPE_STS_NORTH_ALL_0 | 0 | GPI_GPE_EN_NORTH_ALL_0 |
| 1 | GPI_GPE_STS_NORTH_ALL_1 | 1 | GPI_GPE_EN_NORTH_ALL_1 |
| 2 | GPI_GPE_STS_SOUTH_DFX_0 | 2 | GPI_GPE_EN_SOUTH_DFX_0 |
| 3 | GPI_GPE_STS_SOUTH_GROUP0_0 | 3 | GPI_GPE_EN_SOUTH_GROUP0_0 |
| 4 | GPI_GPE_STS_SOUTH_GROUP0_1 | 4 | GPI_GPE_EN_SOUTH_GROUP0_1 |
| 5 | GPI_GPE_STS_SOUTH_GROUP1_0 | 5 | GPI_GPE_EN_SOUTH_GROUP1_0 |
| 6 | GPI_GPE_STS_SOUTH_GROUP1_1 | 6 | GPI_GPE_EN_SOUTH_GROUP1_1 |
| 7 | Reserved | 7 | Reserved |

| GPE Group Name | Three GPIO GPE Status Group and Three GPIO Enable Group Mapped to GPE Group | | |
|---|---|---|---|
| GPE0_DW0 | 0 | GPI_GPE_STS_NORTH_ALL_0 | GPE_GPE_EN_NORTH_ALL_0 |
| GPE0_DW1 | 1 | GPI_GPE_STS_SOUTH_ALL_1 | GPI_GPE_EN_SOUTH_ALL_1 |
| GPE0_DW2 | 2 | GPI_GPE_STS_SOUTH_DFX_0 | GPI_GPE_EN_SOUTH_DFX_0 |

The SoC Power Management Controller (PMC) has its own set of GPE0 status/enable Dwords (GPE0_DW3). See Chapter 16, "Power Management Controller (PMC)".

The SoC PMC is a proxy for the GPIO GPE0 Enable and Status registers. When accessed from Host I/O Space, software writes/reads all of the GPE0 registers directly to/from the PMC. See Table 8-5 on page 342. The GPE0 registers are described in Chapter 58, "Power Management Controller - B0, D31, F2".

When applicable the SoC PMC forwards these writes/reads to GPIO controller based on the mapping path from PMC to GPIO. The GPIO mapping path is configured by the SoC in the two GPIO Sideband registers, offset 10h of Sideband Ports C2h and C5h. These Sideband Ports are accessible in Host Memory Space using SBREG_BAR.

**Table 8-5. Access to the ACPI General Purpose Event Registers**

| Host I/O Space PCU B0:D31:F2 ABASE+Offset (hex) | ACPI General Purpose Event 0 (GPE0) | | Customer GPIO Group |
|---|---|---|---|
| 80 | Status [31:0] | GPE0_DW0 | GPI_GPE_STS_NORTH_ALL_0 |
| 90 | Enable [31:0] | | GPE_GPE_EN_NORTH_ALL_0 |
| 84 | Status [63:32] | GPE0_DW1 | GPI_GPE_STS_NORTH_ALL_1 |
| 94 | Enable [63:32] | | GPI_GPE_EN_NORTH_ALL_1 |
| 88 | Status [95:64] | GPE0_DW2 | GPI_GPE_STS_SOUTH_DFX_0 |
| 98 | Enable [95:64] | | GPI_GPE_EN_SOUTH_DFX_0 |
| 8C | Status [127:96] | GPE0_DW3 | Reserved |
| 9C | Enable [127:96] | | |

The programming steps to map the GPIOs to the GPE0 DWords are as follows:

- Program GPIO Miscellaneous Configuration Register (MISCCFG) located at SBREG_BAR offset 0xC50010h and 0xC20010h bit 8-19 to assign a specific GPIO Group to the ACPI GPE0.
  - Program a specific GPIO Group to GPE0[31:0] as GPE0_DW0
  - Program a specific GPIO Group to GPE0[63:32] as GPE0_DW1
  - Program a specific GPIO Group to GPE0[95:64] as GPE0_DW2
- Program PMC GPIO Configuration Register (GPIO_CFG) at PCI_CFG_PWRMBASE offset 120h to assign a specific GPIO register to the ACPI GPE0.
  - Program a specific GPIO Group to GPE0[31:0] as GPE0_DW0
  - Program a specific GPIO Group to GPE0[63:32] as GPE0_DW1
  - Program a specific GPIO Group to GPE0[95:64] as GPE0_DW2
- Enable 96 GPE available to PMC by writing to PMC. PMC forwards the writes to the GPIO based on the mapping in first step.
  - Program GPE0_EN_31_0 at I/O space 0x90h
  - Program GPE0_EN_63_32 at I/O space 0x94h
  - Program GPE0_EN_95_64 at I/O space 0x98h
- Check GPE0 status by reading from PMC. PMC forwards the reads to the GPIO based on mapping in first step.
  - Read GPE0_STS_31_0 at I/O space 0x80h
  - Read GPE0_STS_63_32 at I/O space 0x84h
  - Read GPE0_STS_95-64 at I/O space 0x88h

For individual GPIO GPE event enable and status detail, please refer to Chapter 26, "Customer General-Purpose I/O (GPIO)".

## 8.6 Message Signaled Interrupt (MSI and MSI-X)

Message Signaled Interrupt (MSI) is an optional feature that enables a device function to request service by using PCI Dword memory write transaction. MSI-X defines a separate optional extension to the basic MSI functionality. During device configuration, each capable PCI function is allocated one or more vectors and the memory-mapped location to write the interrupt messages. Compared to MSI, MSI-X supports a larger number of vectors function, the ability for the OS to program independent address and data values for each vector. Each function in the PCIe Endpoint (EP) supports the MSI capability. MSI-X is supported only for the physical function.

Once an MSI has been generated using MSI-X, its characteristics are similar to an MSI what was generated using an MSI capability record. However, MSI-X supports per-vector masking that is optional for MSI. MSI-X also supports a Function Mask bit. When MSI-X enabled, EP generates an interrupt using a PCIe posted memory write transaction. The address and data of the transaction are determined by the system and programmed in the PCI MSI-X table entry.

## 8.7 I/O APIC Input Mapping

The SoC has an integrated I/O APIC which supports 0-119 APIC interrupts. Each interrupt has its own unique vector assigned by the software. The interrupt vectors are mapped as shown in Table 8-6.

**Table 8-6. I/O APIC Input Mapping (Sheet 1 of 2)**

| I/O APIC Input | Interrupts Routed to This I/O APIC Input | Note |
|---|---|---|
| IRQ0 | Cascaded from Master 8259 | |
| IRQ1 | Option for SERIRQ (1) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ2 | 8254 Counter 0, HPET Timer 0 | 1, 2, 3 |
| IRQ3 | Option for SERIRQ (3) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ4 | Option for SERIRQ (4) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ5 | Option for SERIRQ (5) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ6 | Option for SERIRQ (6) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ7 | Option for SERIRQ (7) or other configurable source including internal ACPI devices, GPIO | 5, 6 |
| IRQ8 | HPET Timer 1 (Legacy mode if GEN_CFG.LEG_RT_CNF is set), RTC (if GEN_CFG.LEG_RT_CNF is not set) | 1, 2, 3 |
| IRQ9 | Option for SERIRQ (9) or other configurable source including internal ACPI devices, SCI, GPIO | 5, 6 |
| IRQ10 | Option for SERIRQ (10) or other configurable source including internal ACPI device, SCI, GPIO | 5, 6 |
| IRQ11 | Option for SERIRQ (11) or other configurable source including internal ACPI devices, SCI, GPIO or HPET Timer 2 | 4 |
| IRQ12 | Option for SERIRQ (12) or other configurable source including internal ACPI devices and GPIO or HPET Timer 3 | 5, 6 |
| IRQ13 | Option for SERIRQ (13) or other configurable source including internal ACPI devices and GPIO | 5, 6 |
| IRQ14 | Option for SERIRQ (14) or other configurable source including internal ACPI devices and GPIO | 5, 6 |

## Table 8-6. I/O APIC Input Mapping (Sheet 2 of 2)

| I/O APIC Input | Interrupts Routed to This I/O APIC Input | Note |
|---|---|---|
| IRQ15 | Option for SERIRQ (15) or other configurable source including internal ACPI devices and GPIO | 5, 6 |
| IRQ16 | PIRQA, or other configurable source including internal ACPI devices and GPIO | 5 |
| IRQ17 | PIRQB, or other configurable source including internal ACPI devices and GPIO | 5 |
| IRQ18 | PIRQC, or other configurable source including internal ACPI devices and GPIO | 5 |
| IRQ19 | PIRQD, or other configurable source including internal ACPI devices and GPIO | 5 |
| IRQ20 | PIRQE, or other configurable source including internal ACPI device, GPIO, SCI, and HPET | 5 |
| IRQ21 | PIRQF, or other configurable source including internal ACPI devices, GPIO, SCI, and HPET | 5 |
| IRQ22 | PIRQG, or other configurable source including internal ACPI devices, GPIO, SCI, and HPET | 5 |
| IRQ23 | PIRQH, or other configurable source including internal ACPI devices and SCI, or GPIO | 5 |
| IRQ24:IRQ 118 | Other configurable source including internal ACPI devices and GPIO | 5 |
| IRQ119 | GPIO | 5 |

***Notes:***
1. General Configuration Register (GEN_CFG) located at the memory space address FED0_0010h has bit 1 as is the Legacy Rout Enable (LEG_RT_CNF).
2. When LEG_RT_CNF is set, the HPET Timer 0 Config and Capabilities (TMR0_CNF_CAP) register and Timer 1 Configuration and Capabilities (TMR1_CNF_CAP) register and TIMER1_INT_ROUT_CNF bits have no impact for timers 0 and 1. TIMER0_CNF_CAP and TIMER1_CNF_CAP are located at the memory-space addresses FED0_0100h and FED0_0120h, respectively.
3. When LEG_RT_CNF is cleared, each of the three HPET timers has its own routing control. The interrupts can be routed to various interrupts in the I/O APIC. See detail in TMRx_CNF_CAP registers.
4. HPET Timer 2 is routed to the APIC as per the routing in the HPET T2C register located at the memory-space addresses FED0_0140h.
5. When programming the polarity of internal interrupt sources on the APIC, interrupts 0 through 15 receives active -high internal interrupt sources; interrupt 16 through 23 receive active-low internal interrupt sources; interrupts 24 through 119 receive active -high internal interrupt sources.
6. SERIRQ is MUXed with GPIO [11] for assertion by external devices. Interrupt SERIRQ is not exposed if the signal is configured as GPIO.

The I/O APIC has a Redirection Table (RT) with an entry for each interrupt source. Each RT entry is individually programmed for trigger mode (edge or level), vector number, and destination processor(s). The interrupt is reported to the appropriate local APIC(s).

For more information about the I/O APIC, see Chapter 21, "PIC and I/O APIC Controllers."

## 8.8 8259 PIC Input Mapping

The interrupts that can be routed to the input of the integrated 8259 PIC are shown in Table 8-7. The 8259 PIC registers are described in Chapter 21, "PIC and I/O APIC Controllers."

### Table 8-7. 8259 PIC Input Mapping

| I/O PIC Input | Master or Slave 8259 PIC Input | Interrupts Routed to This PIC Input | Note |
|---|---|---|---|
| IRQ0 | Master IRQ0 | HPET Timer 0 (Legacy mode if GGEN_CFG.LEG_RT_CNF is set)<br>8254 Timer 0 (if GGEN_CFG.LEG_RT_CNF is not set) | 1 |
| IRQ1 | Master IRQ1 | Option for SERIRQ (1) or other configurable source including SMB/eMMC | 1, 6 |
| IRQ2 | Master IRQ2 | Cascade from Slave 8259 | 1, 2 |
| IRQ3 | Master IRQ3 | Option for SERIRQ (3) or other configurable source including PIRQx and internal ACPI devices | 1, 3, 4, 6 |
| IRQ4 | Master IRQ4 | Option for SERIRQ (4) or other configurable source including PIRQx and internal ACPI devices | 1, 3, 4, 6 |
| IRQ5 | Master IRQ5 | Option for SERIRQ (5) or other configurable source including PIRQx and internal ACPI devices | 1, 3, 4, 6 |
| IRQ6 | Master IRQ6 | Option for SERIRQ (6) or other configurable source including PIRQx and internal ACPI devices | 1, 3, 4, 6 |
| IRQ7 | Master IRQ7 | Option for SERIRQ (7) or other configurable source including PIRQx and internal ACPI devices | 1, 3, 4, 6 |
| IRQ8 | Slave IRQ0 | HPET Timer 1 (Legacy mode if GEN_CFG.LEG_RT_CNF is set)<br>RTC (if GEN_CFG.LEG_RT_CNF is not set) | 1 |
| IRQ9 | Slave IRQ1 | Option for SERIRQ (9) or other configurable source including PIRQx and internal ACPI devices, SCI | 1, 3, 4, 6 |
| IRQ10 | Slave IRQ2 | Option for SERIRQ (10) or other configurable source including PIRQx and internal ACPI device, SCI | 1, 3, 4, 5 |
| IRQ11 | Slave IRQ3 | Option for SERIRQ (11) or other configurable source including PIRQx and internal ACPI devices, SCI, or HPET Timer 2 | 1, 3, 4, 5, 6 |
| IRQ12 | Slave IRQ4 | Option for SERIRQ (12) or other configurable source including PIRQx and internal ACPI devices, or HPET Timer 3 | 1, 3, 4, 6 |
| IRQ13 | Slave IRQ5 | Option for SERIRQ (13) or other configurable source including internal ACPI devices | 1, 6 |
| IRQ14 | Slave IRQ6 | Option for SERIRQ (14) or other configurable source including PIRQx, GPIO and internal ACPI devices | 1, 3, 4, 6 |
| IRQ15 | Slave IRQ7 | Option for SERIRQ (15) or other configurable source including PIRQx, GPIO and internal ACPI devices | 1, 3, 4, 6 |

***Notes:***
1. Interrupts can individually be programmed to be edge or level, except for IRQ0, IRQ1, IRQ2, IRQ8, which are always default to edge.
2. The slave 8259 controller is cascaded onto the master 8259 controller through the master controller interrupt input IRQ2.
3. If an interrupt is used for PCI IRQA - IRQH, SCI, it should not be used for ISA-style interrupts (vis SERIRQ).
4. PCI interrupts are mapped to IRQ3, 4, 5, 6, 7, 9, 10, 11, 12, 14, or 15. It can be programmed by PIRQx Routing Control Register at 0xD0003100h -3107h.
5. If IRQ11/IRQ12 used for HPET#2/HPET#3, software should ensure IRQ11 is not shared with any other devices to guarantee the proper operation of HPET#2/HPET#3. The hardware does not prevent sharing of IRQ11/IRQ12.
6. SERIRQ is MUXed with GPIO [11] for assertion by external devices. Interrupt SERIRQ is not exposed if the signal is configured as GPIO.

## 8.9 Device Interrupt-Generating Capabilities

Integrated devices that generate interrupts are shown in Table 8-8.
Interrupt-generating details of each is located in its individual functional chapter.

**Table 8-8. Device Interrupt / Event Sources**

| Agent | IRQ to ITSS | INTx to ITSS | MSI to CPU | NMI to ITSS | SCI to PMC |
|---|---|---|---|---|---|
| ME.HECI[1-3]<br>IE.HECI[1-3] | - | INT[A-D] | YES | YES, from HECI1 only | YES |
| ME PTIO KT and IDER<br>IE PTIO KT and IDER | - | INT[A-D] | YES | - | - |
| eMMC | IRQ[1-31] | - | - | - | YES |
| GPIO | YES | - | - | YES | YES |
| HSUART | | INT[A-D] | YES | - | - |
| RCEC | | INT[A-D] | YES | - | - |
| GLREG | - | - | - | YES | YES |
| IOMMU | - | - | YES | - | - |
| HPET in ITSS | IRQ8 | - | - | - | - |
| ITSS | YES | - | - | YES | YES |
| LAN | - | INT[A-D] | YES | - | - |
| LPC | IRQ[0-15] | INT[A-D] | - | YES | YES |
| Intel® QAT | - | INT[A] | YES | - | - |
| Intel® Trace Hub | - | INT[A-D] | YES | - | - |
| PCIe Root Port | - | INT[A-D] | YES | - | YES |
| PMC | YES | - | - | - | YES |
| RTC | IRQ8 | - | - | - | YES |
| SATA | - | INT[A-D] | YES | - | - |
| SMBus Host | - | INT[A] | YES | - | - |
| SMBus Legacy | NO | INT[A-D] | - | NO | YES |
| SPI | - | - | - | - | YES |
| VRP[0-2] | - | INT[A-D] | YES | - | - |
| USB Combo | - | INT[A] | YES | - | - |

§ §

# 9 Memory Controller

## 9.1 Introduction

The Memory Controller is a dual channel DDR4 Memory Controller. The SoC supports a 40 or a 72 bit wide DDR4 interface per channel, 32 or 64 bits of data and 4 or 8 bits of ECC, and it does support both ECC and non-ECC DIMMs.

**Figure 9-1. What is Covered in This Chapter**

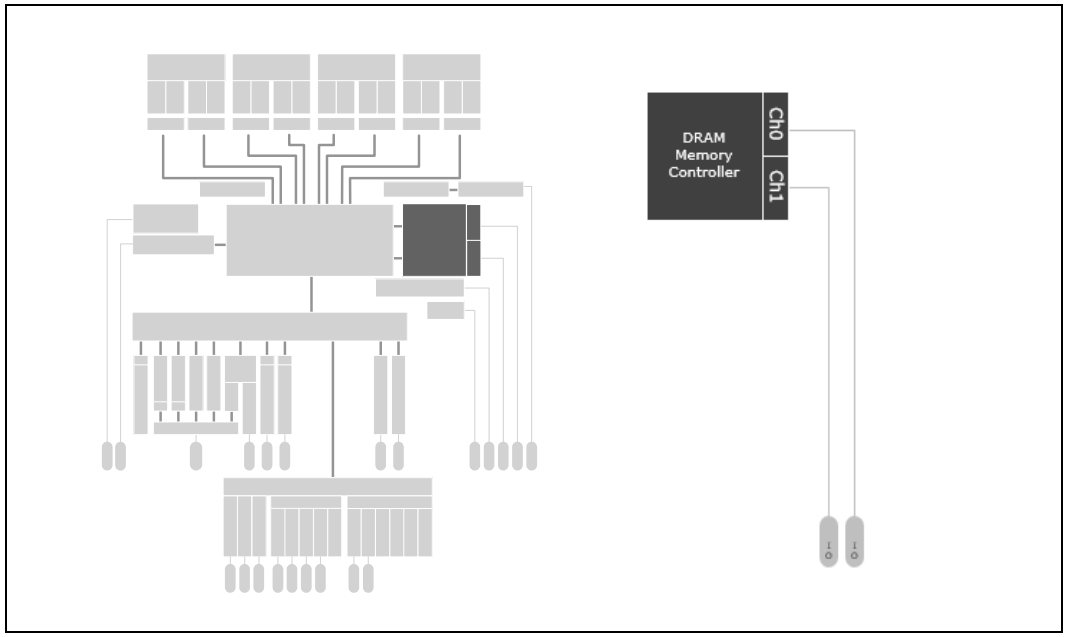



**Table 9-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| JESD79-4 | - | July 2012 | JEDEC DDR4 SDRAM Specification |

## 9.2 Signal Descriptions

The signal descriptions are shown in Table 8-3. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 511. The Direction/Type column of Table 8-3 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.
  Differential = Differential signal pair. Two pins: Positive; Negative.

*Note:* The SoC memory signal pins have SoC-internal termination designed to conform to the DDR4 (JESD79-4) standard. The DDR0_ALERT_N_PAR_ERR_N and DDR1_ALERT_N_PAR_ERR_N SoC input pins do not have any SoC-internal termination. Here the signal termination for each of these signals must be provided by the board design.

**Table 9-2. Signal Names and Descriptions (Sheet 1 of 5)**

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| **Memory Channel 0** | | | |
| DDR0_DQ[63:0] | I,O | No | **Data Lines:** Bi-directional 64-bit data bus, driven on writes, received on reads.<br>***Note:*** In 32-bit mode, the lower 32 bits must be used. |
| DDR0_ECC[7:0] | I,O | No | **ECC Data Lines:** Bi-directional ECC data extending the data bus, driven on writes, received on reads.<br><br>***Notes:***<br>1. If ECC is used 32 and 64 bit data width requires 8 bits of ECC.<br>2. The ECC SDRAM device must be of the same type and width as the SDRAM devices used for the data bits. |
| DDR0_DQS_DN[17:0]<br>DDR0_DQS_DP[17:0] | I,O, Differential | No | **Data Strobes:** During writes, driven by Cell Delay Variation (CDV) offset so as to be centered in the data phase. During reads, driven by memory devices edge aligned with data. The following list matches the data strobe in the data group.<br><br>DQS_DP/DN[7]: DQ[63:56]<br>DQS_DP/DN[6]: DQ[55:48]<br>DQS_DP/DN[5]: DQ[47:40]<br>DQS_DP/DN[4]: DQ[39:32]<br>DQS_DP/DN[3]: DQ[31:24]<br>DQS_DP/DN[2]: DQ[23:16]<br>DQS_DP/DN[1]: DQ[15:8]<br>DQS_DP/DN[0]: DQ[7:0]<br><br>If ECC is supported then DQS_DP/DN[8]: ECC[7:0].<br>For x4 devices, strobe is divided to be high nibble strobe and low nibble strobe.<br>***Note:*** In 32-bit mode, the lower DQ must be used. |
| DDR0_MA[13:0] | O | No | **Multiplexed Address:** Provides multiplexed row and column address to memory. Provides the row address for Active commands and the column address and Auto Precharge bit for Read/Write commands to select one location out of the memory array in the respective bank. A10 is sampled during a Precharge command to determine whether the Precharge applies to one bank (A10 LOW) or all banks (A10 HIGH). If only one bank is to be precharged, the bank is selected by BA0 - BA2. The address outputs also provide the op-code during MRS or EMRS commands. |

## Table 9-2. Signal Names and Descriptions (Sheet 2 of 5)

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| DDR0_MA14_WE_N | O | No | **DDR0_WE_N:** When DDR0_ACT_N_MA[15] signal is high, otherwise DDR0_MA[14]. |
| DDR0_MA15_CAS_N | O | No | **DDR0_CAS_N:** When DDR0_ACT_N_MA[15] signal is high, otherwise DDR0_MA[15]. |
| DDR0_MA16_RAS_N | O | No | **DDR0_RAS_N:** When DDR0_ACT_N_MA[15] signal is high, otherwise DDR0_MA[16]. |
| DDR0_MA17 | O | No | **Address Lines:** Provide the row address for ACTIVATE commands and column address for Read/Write commands. This address line is only defined for the x4 configuration. |
| DDR0_ACT_N_MA[15] | O | No | **Activation Command ACT_N:** Defines the Activation command being entered along with CS_N. |
| DDR0_CLK_DN[3:0]<br>DDR0_CLK_DP[3:0] | O, Differential | No | **Differential DDR Clock:** All address and control input signals are sampled on the crossing of the positive edge of CK_P and negative edge of CK_N. Output (read) data is referenced to the crossings of CK_P and CK_N (both directions of crossing). |
| **Memory Channel 0 (continued)** | | | |
| DDR0_CKE[3:0] | O | No | **Clock Enable:** (active high) CKE is used for power control of the DRAM devices.<br>CKE HIGH activates and CKE LOW deactivates DRAM device internal clock signals and device input buffers and output drivers.<br>These are output signals from the SoC. |
| DDR0_CS_N[3:0] | O | No | **Chip Select:** (active low) These signals determine whether a command is valid in a given cycle for the devices connected to it. All commands are masked when CS_N is registered HIGH. CS_N provides for external rank selection on systems with multiple ranks. CS_N is considered part of the command code. |
| DDR0_C0 | O | No | **Chip ID:** Not supported by the SoC. Board designs shall treat this pin as a No-Connect (NC). |
| DDR0_BA[1:0] | O | No | **Bank Address:** Defines the bank which is the destination for the current Activate, read, write or precharge command. |
| DDR0_BG0_BA[2] | O | No | Bank group address. |
| DDR0_BG1_MA[14] | O | No | Bank group address. |
| DDR0_ODT[3:0] | O | No | **On-Die Termination Enable:** (active high) ODT (registered HIGH) enables termination resistance internal to the DDR device SDRAM. When the ODT feature is enabled, it is dynamically enabled for the receiver of the data. The SoC does this internally for read data returning from the DRAM devices. For write data to the DRAM devices, the DDR0_ODT[] pins are asserted to enable ODT w/in the DRAM devices themselves.<br>Because ODT consumes power, when the feature is enabled, it is controlled dynamically by the SoC.<br>ODT impacts the DQ and DQS. |
| DDR0_PAR | O | No | Command and address parity |
| DDR0_ALERT_N_PAR_ERR_N | I | No | **Alert:** Command and address parity error flag or write data CRC error flag (Note the SoC does not support CRC). There is no SoC-internal termination to this input signal. This SoC input is the receiver of the active-low DDR4 "ALERT_n" signal bus which is possibly driven by more than one system source. Board-level termination is required. |

## Table 9-2. Signal Names and Descriptions (Sheet 3 of 5)

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| **Memory Channel 1** | | | |
| DDR1_DQ[63:0] | I,O | No | **Data Lines:** Bi-directional 64-bit data bus, driven on writes, received on reads.<br>***Note:*** In 32-bit mode, the lower 32 bits must be used. |
| DDR1_ECC[7:0] | I,O | No | **ECC Data Lines:** Bi-directional ECC data extending the data bus, driven on writes, received on reads.<br><br>***Notes:***<br>1. If ECC is used 32 and 64 bit data width requires 8 bits of ECC.<br>2. The ECC SDRAM device must be of the same type and width as the SDRAM devices used for the data bits. |
| DDR1_DQS_DP[17:0]<br>DDR1_DQS_DN[17:0] | I,O, Differential | No | **Data Strobes:** During writes, driven by Cell Delay Variation (CDV) offset so as to be centered in the data phase. During reads, driven by memory devices edge aligned with data. The following list matches the data strobe in the data group.<br><br>DQS_DP/DN[7]: DQ[63:56]<br>DQS_DP/DN[6]: DQ[55:48]<br>DQS_DP/DN[5]: DQ[47:40]<br>DQS_DP/DN[4]: DQ[39:32]<br>DQS_DP/DN[3]: DQ[31:24]<br>DQS_DP/DN[2]: DQ[23:16]<br>DQS_DP/DN[1]: DQ[15:8]<br>DQS_DP/DN[0]: DQ[7:0]<br><br>If ECC is supported then DQS_DP/DN[8]: ECC[7:0].<br>DDR4 supports differential strobes only.<br>For x4 devices, strobe is divided to be high nibble strobe and low nibble strobe.<br>***Note:*** In 32-bit mode, the lower DQ must be used. |
| DDR1_MA[13:0] | O | No | **Multiplexed Address:** Provides multiplexed row and column address to memory. Provides the row address for Active commands and the column address and Auto Precharge bit for Read/Write commands to select one location out of the memory array in the respective bank. A10 is sampled during a Precharge command to determine whether the Precharge applies to one bank (A10 LOW) or all banks (A10 HIGH). If only one bank is to be precharged, the bank is selected by BA0 - BA2. The address inputs also provide the op-code during MRS or EMRS commands. |
| DDR1_MA14_WE_N | O | No | **DDR1_WE_N:** When DDR1_ACT_N_MA[15] signal is high, otherwise DDR1_MA[14] |
| DDR1_MA15_CAS_N | O | No | **DDR1_CAS_N:** When DDR1_ACT_N_MA[15] signal is high, otherwise DDR1_MA[15] |
| DDR1_MA16_RAS_N | O | No | **DDR1_CAS_N:** When DDR1_ACT_N_MA[15] signal is high, otherwise DDR1_MA[15] |
| DDR1_MA17 | O | No | **Address Lines:** Provide the row address for ACTIVATE commands and column address for Read/Write commands. This address line is only defined for the x4 configuration. |
| DDR1_ACT_N_MA[15] | O | No | **Activation Command Input:** ACT_N defines the Activation command being entered along with CS_N |
| DDR1_CLK_DP[3:0]<br>DDR1_CLK_DN[3:0] | O, Differential | No | **Differential DDR Clock:** All address and control input signals are sampled on the crossing of the positive edge of CK_P and negative edge of CK_N. Output (read) data is referenced to the crossings of CK_P and CK_N (both directions of crossing). |

## Table 9-2. Signal Names and Descriptions (Sheet 4 of 5)

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| DDR1_CKE[3:0] | O | No | **Clock Enable:** (active high) CKE is used for power control of the DRAM devices.<br>CKE HIGH activates and CKE LOW deactivates DRAM device internal clock signals and device input buffers and output drivers.<br>These are output signals from the SoC. |

## Table 9-2. Signal Names and Descriptions (Sheet 5 of 5)

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| **Memory Channel 1 (continued)** | | | |
| DDR1_CS_N[3:0] | O | No | **Chip Select:** (active low) These signals determine whether a command is valid in a given cycle for the devices connected to it. All commands are masked when CS_N is registered HIGH. CS_N provides for external rank selection on systems with multiple ranks. CS_N is considered part of the command code. |
| DDR1_C0 | O | No | **Chip ID:** Not supported by the SoC. Board designs shall treat this pin as a No-Connect (NC). |
| DDR1_BA[0:1] | O | No | **Bank Address:** Defines the bank which is the destination for the current Activate, read, write or precharge command. |
| DDR1_BG0_BA[2] | O | No | Bank group address. |
| DDR1_BG1_MA[14] | O | No | Bank group address. |
| DDR1_ODT[3:0] | O | No | **On-Die Termination Enable:** (active high). ODT (registered HIGH) enables termination resistance internal to the DDR device SDRAM. When the ODT feature is enabled, it is dynamically enabled for the receiver of the data. The SoC does this internally for read data returning from the DRAM devices. For write data to the DRAM devices, the DDR1_ODT[] pins are asserted to enable ODT w/in the DRAM devices themselves. Because ODT consumes power, when the feature is enabled, it is controlled dynamically by the SoC. ODT impacts the DQ and DQS. The ODT pin will be ignored by the DDR devices if the EMR(1) is programmed to disable ODT. One pin per rank. |
| DDR1_PAR | O | No | Command and address parity. |
| DDR1_ALERT_N_PAR_ERR_N | I | No | **Alert:** Command and address parity error flag or write data CRC error flag (Note the SoC does not support CRC). There is no SoC-internal termination to this input signal. This SoC input is the receiver of the active-low DDR4 "ALERT_n" signal bus which is possibly driven by more than one system source. Board-level termination is required. |
| **Signals Common to Both Memory Channels** | | | |
| DDR01_DRAMRST_N | O | No | **DRAM Reset:** Active low Reset to the DDR DIMMs of both Memory Controllers. |
| DDR01_COMP | O | No | The platform board must provide a 1% resistor from this pin to VSS:<br>• 392 Ω for DDR4 |
| ADR_TRIGGER | I,O-OD | Yes | **Asynchronous DRAM Self-Refresh (ADR) Trigger:** Chapter 16, "Power Management Controller (PMC)" |
| ADR_COMPLETE | I,O | Yes | **Asynchronous DRAM Self-Refresh (ADR) Complete:** Chapter 16, "Power Management Controller (PMC)" |

## 9.3 Feature List

- Up to two 32/40 bit [1] or 64/72 bit wide memory channels. Up to two DIMMs per channel
- Single-rank or dual-rank per DIMM module
- ECC [3] and non-ECC support
- DDR4 Low Voltage version operating at 1.2V
  - 1600, 1866 [2], 2133, and 2400 MT/s
  - x8 and x16 width for unbuffered memory (UDIMM, SO-DIMM & memory down)
  - x4 and x8 width devices for RDIMM
  - 4 Gb, 8 Gb, and 16 Gb technology devices
  - Minimum memory is:
    32 bit [1] wide memory channel: 1GB (4 Gb, 1 rank, x16 width)
    64 bit wide memory channel: 2GB (4 Gb, 1 rank, x16 width)
  - Maximum memory single channel is 128GB (16 Gb, x4n width, 4 ranks, 64 bit)
  - Both memory channels maximum is 256GB (128GB with two 64 bit channels)
- SoC memory population rules:
  - Both channels must be 32 bit [1] or 64 bits wide
  - At least 1 memory channel must be populated and enabled.
    Either channel 0 or channel 1 may be disabled
  - Both channels require either unbuffered memory (UDIMM, SO-DIMM or memory down), or registered memory (RDIMM and/or NVRDIMM). Technology mixing is not supported.
  - Rank mixing is allowed but can result in a speed bin loss.

To calculate the capacity of a DIMM, the type of DRAMs used must be understood. The width of the DRAM should be used to find the number of DRAMs needed to make up the data bus. The capacity per rank can then be calculated by multiplying the number of DRAMs by the size of the DRAM, and further divided by eight to convert bits into bytes. The total DIMM capacity is finally calculated by multiplying the rank capacity and the number of ranks on the DIMM.

***Notes:***
1. A0 and A1 step silicon only support 64/72-bit wide memory channels.
2. The silicon design supports 1866 MT/s but is not validated on Intel platforms. Customers are responsible for validation.
3. The ECC SDRAM device must be of the say type and width as the SDRAM devices used for the data bits. This limitation is driven by the DDR4 spec.
   - It is NOT alright to use x16 SDRAM devices for data and a x8 device for ECC.
   - It is NOT alright to use x8 SDRAM devices for data and a x4 device for ECC.
4. Memory based on 16 Gb dual rank technology (including maximum memory configuration of 256GB) is supported by silicon but not validated on Intel platforms.

# 9.4 Memory Controller Power Management Features

## 9.4.1 DRAM Running Average Power Limit (RAPL)

DRAM Running Average Power Limiting (RAPL) allows power limiting for the memory connected to the SoC.

DRAM RAPL provides a way to set power limits on your memory power consumption. This will allow the SoC to dynamically limit max average power, to match its expected power budget. DRAM RAPL Domain only includes the memory connected directly to the SoC. Implementing RAPL across multiple SoC must be implemented in a multi-node management RAPL solution. DRAM RAPL provides a way to set short term and longer term averaging windows for power limits for the SoC DRAM Memory.

## 9.4.2 Dynamic (HW autonomous) Self-refresh DDR4

The Self-Refresh command can be used to retain data in the DDR4 DRAM, even if the rest of the system is powered down. When in the Self-Refresh mode, the DDR4 SDRAM retains data without external clocking.The DDR4 DRAM device has a built-in timer to accommodate Self-Refresh operation. The Self-Refresh-Entry (SRE) Command is defined by having CS_n, RAS_n/A16, CAS_n/A15, and CKE held low with WE_n/A14 and ACT_n high at the rising edge of the clock.

Before issuing the Self-Refresh-Entry command, the DDR4 DRAM must be idle with all bank precharge state with tRP satisfied. 'Idlestate' is defined as all banks are closed (tRP, tDAL, etc. satisfied), no data bursts are in progress, CKE is high, and all timings from previous operations are satisfied (tMRD, tMOD, tRFC, tZQinit, tZQoper, tZQCS, etc.). Deselect command must be registered on last positive clock edge before issuing Self Refresh Entry command. Once the Self Refresh Entry command is registered, Deselect command must also be registered at the next positive clock edge. Once the Self-Refresh Entry command is registered, CKE must be held low to keep the device in Self-Refresh mode. DRAM automatically disables ODT termination and set Hi-Z as termination state regardless of ODT pin and RTT_PARK set when it enters in Self-Refresh mode.

Upon exiting Self-Refresh, DRAM automatically enables ODT termination and set RTT_PARK asynchronously during tXSDLL when RTT_PARK is enabled. During normal operation (DLL on) the DLL is automatically disabled upon entering Self-Refresh and is automatically enabled (including a DLL-Reset) upon exiting Self-Refresh.

When the DDR4 SDRAM has entered Self-Refresh mode, all of the external control signals, except CKE and RESET_n, are "don't care." For proper Self-Refresh operation, all power supply and reference pins (VDD, VDDQ, VSS, VSSQ, VPP, and VRefCA) must be at valid levels. DRAM internal VrefDQ generator circuitry may remain ON or turned OFF depending on DRAM design. If DRAM internal VrefDQ circuitry is turned OFF in self refresh, when DRAM exits from self refresh state, it ensures that VrefDQ generator circuitry is powered up and stable within tXS period. First Write operation or first Write Leveling Activity may not occur earlier than tXS after exit from Self Refresh. The DRAM initiates a minimum of one Refresh command internally within tCKE period once it enters Self-Refresh mode.

The clock is internally disabled during Self-Refresh Operation to save power. The minimum time that the DDR4 SDRAM must remain in Self-Refresh mode is tCKESR. The user may change the external clock frequency or halt the external clock tCKSRE after Self- Refresh entry is registered, however, the clock tCKESR must be restarted and stable before the device can exit Self-Refresh operation.

*Note:* Package C6 (PC6) only helps to save power when DDR dynamic self-refresh feature is on.

### 9.4.3 Unsupported Features and Restrictions

- DDR3 and DDR3L DRAM technology is not support on this product.
- Technology to minimize DDR pass-gate charge migration issues.
- DDR4 data bus widths other than 32/40 bits or 64/72 bits are not supported.
- 2-Gb Device Density (or lower).
- Dual Channel: Both channels do not have to be populated identically
  - Same voltage.
  - ECC versus non-ECC DIMMs; DIMMs must all be ECC or non-ECC; no mixing allowed.
  - DIMMs must be all either UDIMMs, RDIMMs or NVRDIMM. No DIMM type mixing is permitted.
  - Frequency – BIOS will set the overall Memory frequency to the lowest frequency that is detected during SPD discovery across both channels and all populated DIMMs.
- Partial Self Refresh is not supported.
- Memory mirroring is not supported.
- Memory lockstep is not supported.
- Memory sparing is not supported.
- Memory Hot-Swap is not supported.
- Suspend to Memory (S3) is not supported.
- Single Device Data Correction (SDDC) is not supported.

*Note:*

1. SPD setting impacts memory initialization during the system boot up. Please refer to the *SPD Annex L, Serial Presence Detect (SPD) for DDR4 SDRAM Modules* for details.

### 9.4.4 DDR4 DIMM Types and Population

Registered (RDIMM), unbuffered (UDIMM), or small-outline (SODIMM) ECC modules are supported. ECC RDIMMs and UDIMMs are supported. Single rank ("1R") and dual rank ("2R") DIMMs can be used.

*Note:* Contact your Intel representative for the Harrisonville Platform Snapshot document which contains the most up to date supported memory information.

### 9.4.4.1 DIMM Population Requirements

Either channel may be populated or both. There are no DIMM matching requirements between channels. Both channels, however, must run at the same interface frequency. Each channel may run at different DIMM timings (RAS latency, CAS latency, and so forth).

The platform does not support mixing of DIMM types. Sockets are populated with all RDIMMs, all UDIMMs, or all SODIMMs. All DIMMs must be DDR4 DIMMs.

DIMMs with different timing parameters can be installed on different slots within the same channel, but only timings that support the slowest DIMM will be applied, i.e., the faster DIMM in a channel will run at the lower speed.

No mixing of x4 and x8 RDIMMs is allowed within a channel. It is allowed when in different channels. Mixing of single rank (SR) and dual rank (DR) DIMMs is allowed but there may be a speed bin loss.

Each populated channel must be x4 or x8 or x16 only (no width mixing within a channel).

These are summarized in Table 9-3.

Regardless of RAS mode, the requirements for DIMM populating within a System and a channel provided in the System/Channel Level DIMM Population Requirements sections in the Product Design Guide must be met at all times.

**Table 9-3. DIMM Configurations, Two DIMM Socket per Channel**

| Style | 1N/2N | DIMM1 | DIMM0 |
|---|---|---|---|
| RDIMM | 1N | empty [1] | SRx4, SRx8, DRx4, DRx8 |
| | 1N | SRx4 | SRx4 |
| | 1N | SRx8 | SRx8 |
| | 1N | DRx4 | DRx4 |
| | 1N | DRx8 | DRx8 |
| UDIMM | 2N | empty | SRx8, DRx8 |
| | 2N | SRx8 | SRx8 |
| | 2N | DRx8 | DRx8 |
| SODIMM | 2N | empty | SRx8, SRx16, DRx8 |
| | 2N | SRx8 | SRx8 |
| | 2N | SRx16 | SRx16 |
| | 2N | DRx8 | DRx8 |

***Notes:***
1. For designs using the T-topology in a two socket configuration with only one DIMM slot being populated it is recommended to populate DIMM0 first, although either DIMM0 or DIMM1 may be populated but DIMM1 only population is not fully validated by Intel. For daisy chain designs customers need to populate the further connector to decrease the risk of reflections that impacts signal integrity.

# 9.5 Memory Controller RAS Features

## 9.5.1 Data Scrambler

Data scrambling is a general technique to reduce supply noise and improve signal integrity by XORing data and ECC with a pseudo random sequence. The pseudo random sequence has two important effects relative to power delivery. Across the 72 data/ECC bits of the bus, it drives very close to 50% "1's" and 50% "0's" in every cycle. This eliminates the previous worst case where all bits of the bus would simultaneously drive high or low.

## 9.5.2 Patrol Scrub

Patrol Scrubbing is a process that allows the SoC to correct correctable memory errors detected on a memory module and send the correction to the requester (the original source). When this item is set to Enabled, the SoC will analyze one cache line every 16K cycles, if there is no delay caused by internal processing. By using this method, roughly 64 GB of memory connected to the SoC will be scrubbed every 24 hours. This option can be enabled or disabled.

*Note:* The above feature requires ECC to be enabled.

## 9.5.3 Demand Scrub

Demand Scrubbing is a process that allows the SoC to correct correctable memory errors found on a memory module. When the SoC or I/O issues a demand-read command, and the read data from memory turns out to be a correctable error, the error is corrected and sent to the requester (the original source). Memory is updated as well. This option can be enabled or disabled.

*Note:* The above feature requires ECC to be enabled.

## 9.5.4 Asynchronous DRAM Refresh (ADR)

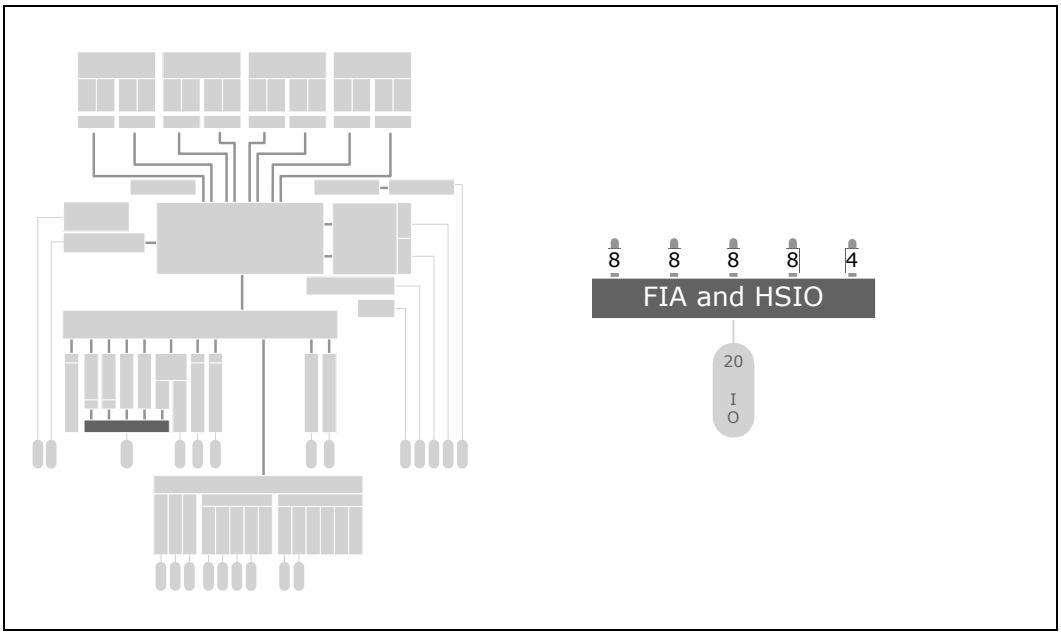
Asynchronous DRAM Refresh is a SoC feature that preserves context data to NVRDIMMs before data loss can occur in the event of an unexpected power loss. Once all inflight data is flushed including CPU cached data, the DRAMs must be placed into a self-refresh state before platform power is lost. Details about ADR are in Section 16.15, "Asynchronous DRAM Refresh (ADR)" on page 547.

§ §

# 10 Flexible I/O Adapter (FIA) Overview

The High Speed I/O (HSIO) lanes are configured by the SoC customer. The product SKU determines how many lanes of PCI Express* Root Ports, SATA ports, and USB 3.0 ports are available. The arrangement of the available lanes is up to the customer. BIOS, via the Intel® Management Engine (Intel® ME), along with Soft Straps and the bifurcation settings of the PCIe* Root Port clusters, configure the lanes.

**Figure 10-1. What is Covered in This Chapter**



## 10.1 Signal Descriptions

The signal descriptions are shown in Table 10-1. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a GPIO signal see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction/Type column of Table 10-1 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- Differential = Differential signal pair. Two pins: Positive; Negative.

*Note:* The HSIO transmit and receive differential pairs for each lane have SoC-internal termination designed to conform to the PCIe*, SATA, or USB 3.0 industry specification for which the lane is configured.

**Table 10-1. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| HSIO_RX_DP[19:0]<br>HSIO_RX_DN[19:0] | I, Differential | Yes | **High-Speed I/O Lane Read Data** - These 20 sets of differential receivers are configured by the customer as PCI Express Root Ports, SATA Ports, and/or USB 3.0 Ports. |
| HSIO_TX_DP[19:0]<br>HSIO_TX_DN[19:0] | O, Differential | Yes | **High-Speed I/O Lane Write Data** - These 20 sets of differential drivers are configured by the customer as PCI Express Root Ports, SATA Ports, and/or USB 3.0 Ports. |
| HSIO_COMPREF_DP<br>HSIO_COMPREF_DN | O, Differential | No | **High-Speed** I/O **Common Lane Compensation Pins** - The platform board must provide a 100 Ω, 1% resistor connected from the DP pin to the DN pin. |
| HSHV_IRCOMP | I,O | No | **HSHV_IRCOMP** - Special compensation resistor to support 3.3V shared by all of the High-Speed I/O Lane interfaces. The platform board must provide a 100-Ω, 1% resistor connected from this pin to VSS. |

## 10.2 Feature List

- Customer-configurable high-speed lane patterns
- Controlled by BIOS, Soft Straps, Intel® ME, and the PCIe bifurcation settings

## 10.3 Architectural Overview

The Flexible I/O Adapter (FIA) is the mechanism that enables the customer to configure the High Speed I/O (HSIO) lanes to fit a particular system need.

Three controller types are involved to determine HSIO usage:

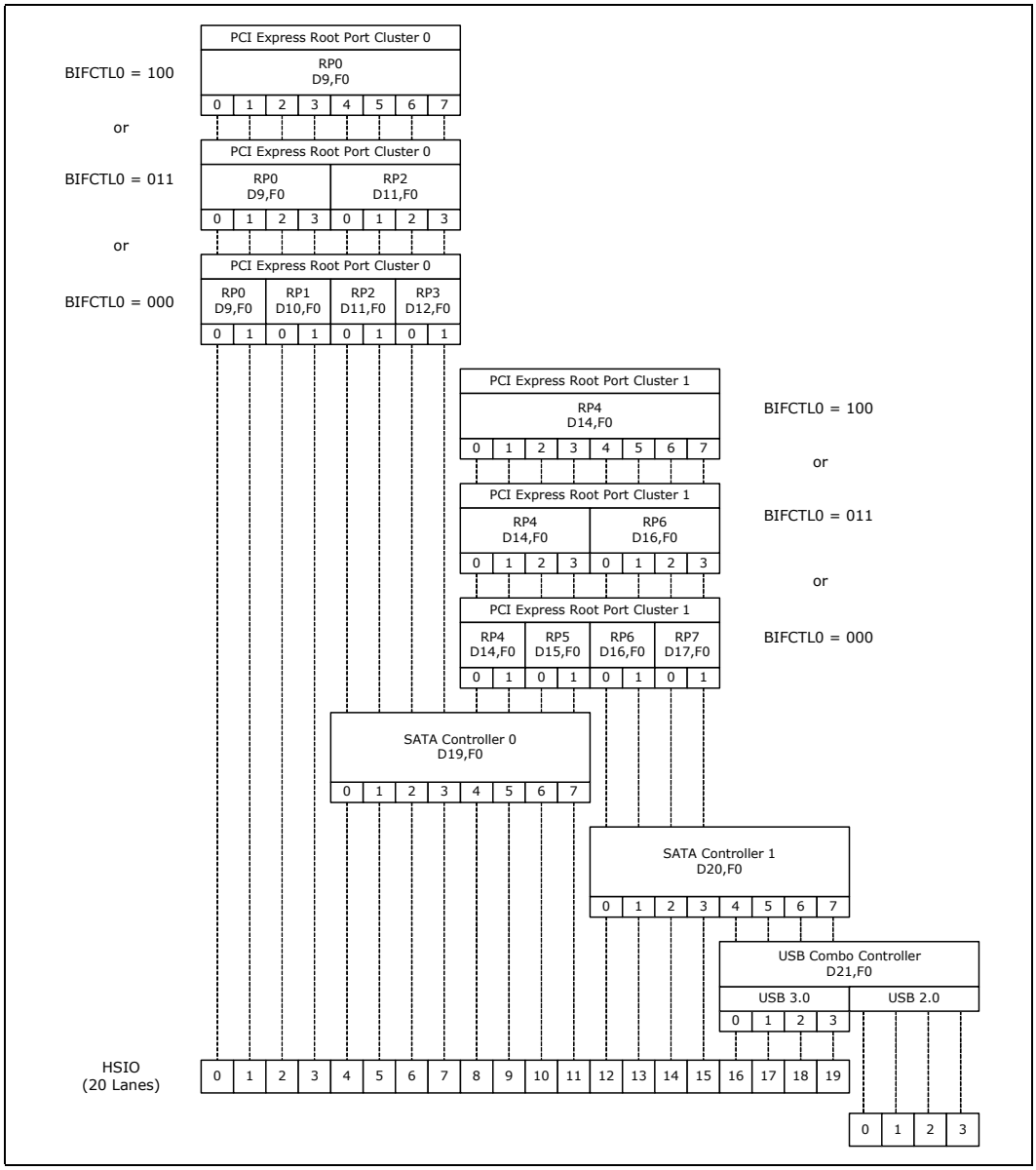
- PCI Express Root Ports
- SATA
- USB 3.0

See Figure 10-2, "FIA and HSIO Structure".

The FIA provides a high degree of configuration flexibility without increasing pin count by allowing a single PHY lane to be statically mapped to different controllers. This provides great flexibility to platform board designers as the number of PCIe, SATA and USB 3.0 ports and their mapping to SoC pin-out can be configured differently for different platforms.

The FIA supports shared clock-control logic that allows the integrated controllers to share a single master clock. The FIA also contributes to dynamic PHY power gating by managing the PHY common-lanes power gating, allowing power gating only when all the controllers permit. This further reduces the idle power of the PHY. The PHY data lanes power gating is managed by the respective controllers.

## Figure 10-2. FIA and HSIO Structure




*Note:* The structures for PCIe RP bifurcation control BIFCTL0 = 001 and 010 are not shown here but are valid configurations. Most use cases are satisfied by what is shown in the figure.

## 10.4 USB 3.0 as a Debug Port

Depending on the product SKU, lanes 19:16 are set to be either SATA or USB 3.0. See Figure 10-2 on page 362. For the debug Intel® Direct Connect Interface (Intel® DCI) to work correctly, the Intel® ME programs the Intel® DCI registers to indicate which lanes are SATA and which are USB 3.0 for the particular SoC product SKU. See Chapter 38, "JTAG and Debug Ports" for Intel® DCI and the SoC JTAG port.

## 10.5 Lane Configuration

All product SKUs are shipped with all controllers enabled for the PCI Express Root Ports, SATA, and USB 3.0 interfaces. The product SKUs are shown in Section 1.1, "Overview." Each product SKU has a limit to the number of HSIO lanes that a customer can use, but this limitation does not specify which physical lanes can and cannot be used. The customer defines the Soft Strap settings (see Chapter 4, "Strapping and Configuration") to connect the I/O lanes of the PCIe Root Port, SATA and USB 3.0 controllers to individual HSIO lanes. Section 10-2, "FIA and HSIO Structure" shows the relationship of the controllers with the HSIO lane numbers. The Soft Straps must define controller connections to no more than the maximum number of HSIO lanes allowed for the particular product SKU. In addition, the PCI Express lane patterns are determined by the values programmed in the PCIe Bifurcation Control Registers described in Section 10.5.2, "PCI Express Lane Configuration."

If the SoC determines that there are more HSIO lanes connected by Soft Straps than allowed by the product SKU, the SoC will configure the HSIO lanes as only four USB 3.0 lanes. In this case, BIOS is still initiated but the system will have only the four USB 3.0 lanes active at the HSIO pins.

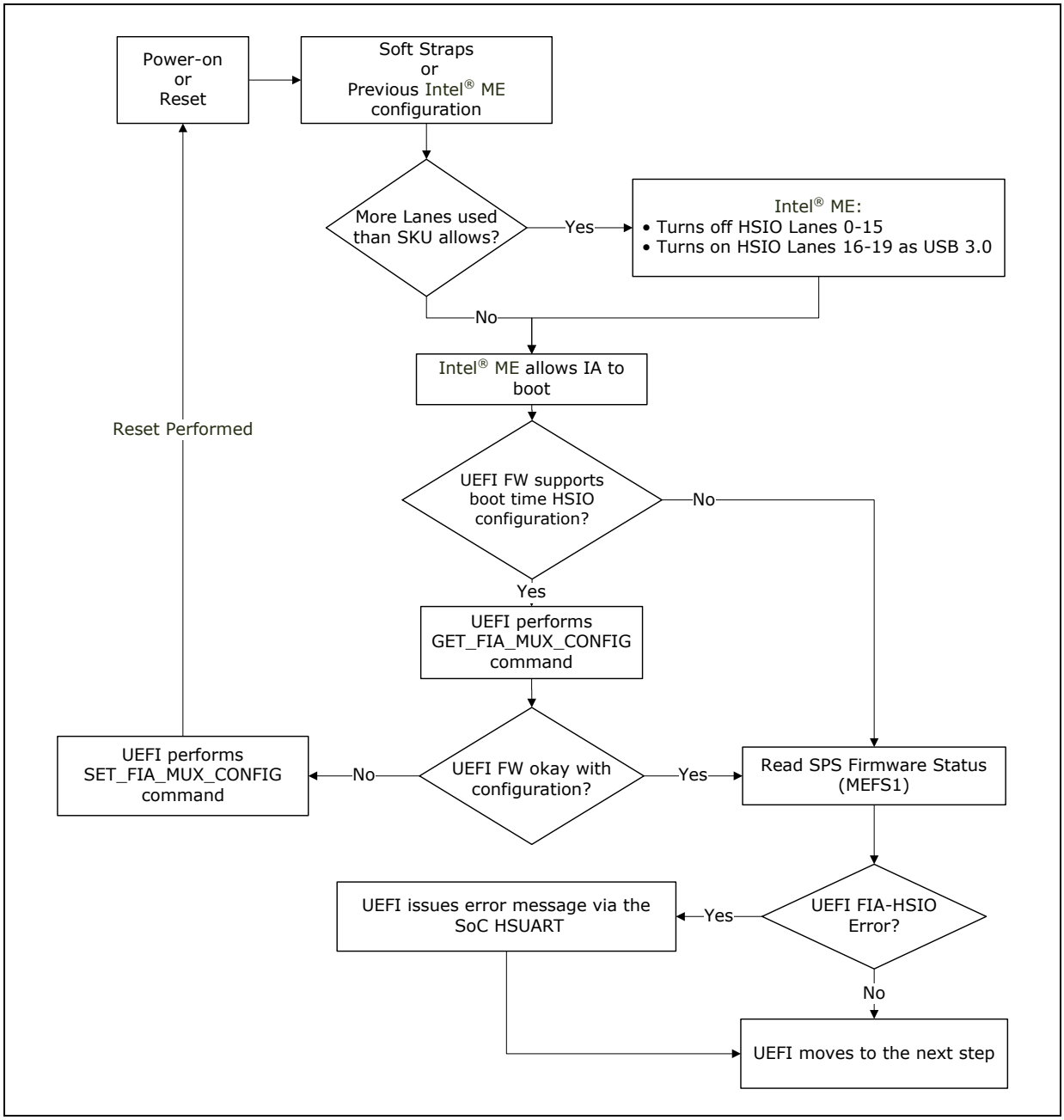
### 10.5.1 Intel® Management Engine and BIOS Roles in Lane Configuration

FIA lane configuration and management are part of the Intel® Server Platform Services (SPS) firmware.

The specification includes a flow diagram for systems with Intel® SPS firmware where the BIOS and Intel® ME determine the lane-assignment validity and confirm that the assignments are within the SoC product SKU definition. This flow is summarized in Figure 10-3, "Intel® ME-BIOS/UEFI Interactions in a System with Intel® SPS Firmware" and addresses three customer use cases:

- The resulting system has a unique integrated firmware image (IFWI) and product-SKU combination for each of the system's boards. This combination is custom-programmed in the SPI device during board manufacturing. Here the SoC Soft Straps provide the HSIO lane assignments.
- A single IFWI for a particular board design is programmed to the SPI device during board manufacturing. The board is designed to accommodate one of a number of SoC product SKUs. Here the SoC Soft Straps in the single IFWI provide the HSIO lane assignments and, if required, the BIOS/UEFI overrides the initial HSIO lane assignments based on the product SKU used on the board.
- A single IFWI for a particular board design is programmed to the SPI Flash memory at manufacturing similar to above, but the desired HSIO assignments cannot be determined until the board is plugged into the chassis and power applied. Here the BIOS/UEFI determines which system I/O cards/controllers are system-connected to the board and attempts to configure the HSIO lanes.

**Figure 10-3. Intel® ME-BIOS/UEFI Interactions in a System with Intel® SPS Firmware**



## 10.5.2 PCI Express Lane Configuration

In addition to lane disabling/enabling, the PCI Express lane patterns are determined by the values programmed in the three-bit Bifurcation Control 0 (BIFCTL0) field of the PCIe Port Bifurcation Control Register (PPBIFCTL_PRIV) registers. There is one physical PPBIFCTL_PRIV register for PCI Express Root Port (RP) Cluster 0 and one for RP Cluster 1. Each of the two 8-lane PCI Express RP Clusters has a PPBIFCTL_PRIV register. Bifurcation of the eight lanes is shown in Table 10-2. The effects of the three unshaded BIFCTL0 decodes are shown in Figure 10-2, "FIA and HSIO Structure." BIFCTL0 = 001 and 010, not shown in the figure, are also valid decodes.

**Table 10-2. Bifurcation Control for PCIe RP Clusters 0 and 1**

| BIFCTL0 [2:0] | Port 0 | Port 1 | Port 2 | Port 3 |
|---|---|---|---|---|
| 000 | X2 | X2 | X2 | X2 |
| 001 | X4 | | X2 | X2 |
| 010 | X2 | X2 | X4 | |
| 011 | X4 | | X4 | |
| 100 | x8 | | | |
| **101** | **Reserved** | | | |
| **110** | | | | |
| **111** | | | | |

There are two access mechanisms for each of the two PCIe Port Bifurcation Control registers. When accessed as a Sideband Register, it is a Read-Write register where the value in BIFCTL0 can be locked once it is set. It can also be accessed through Configuration Space where it is a Read-Only register. The PCIe Port Bifurcation Control registers is called:

- PPBIFCTL_PRIV when accessed as a Sideband Register
- PPBIFCTL when accessed via Configuration Space

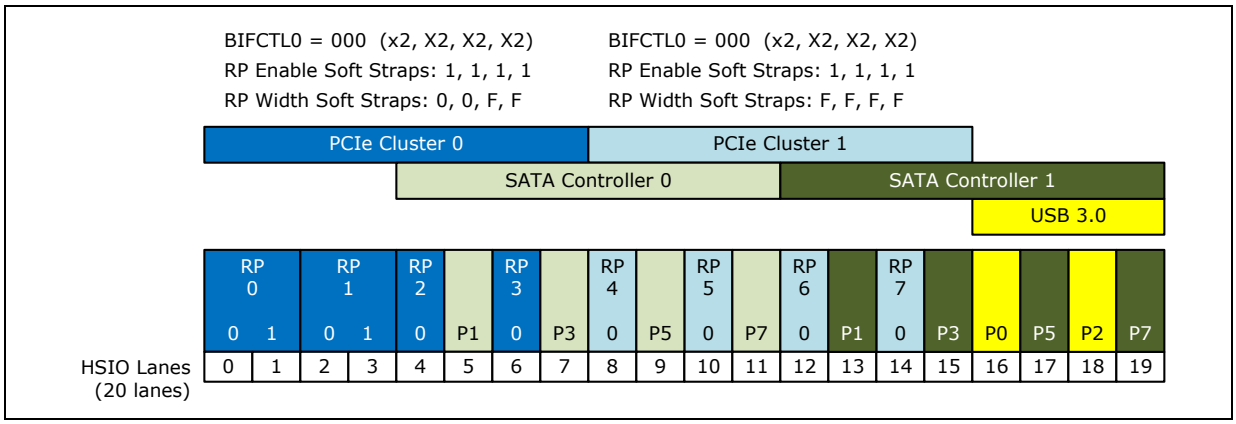
As a Sideband Register, PPBIFCTL_PRIV for PCIe RP Cluster 0 is at Sideband Port B4h offset 10h and at Port B3h offset 10 for RP Cluster 1. These Sideband Registers are accessed in Host Memory-Mapped Space using SBREG_BAR, offsets B4_0010h and B3_0010h respectively.

In Configuration Space, PPBIFCTL is a Device-Specific Register located at offset C0h from B0,D9,F0 for PCIe RP Cluster 0 and is aliased in Configuration Space at F0 for Devices 10, 11, and 12. It represents the lane arrangement of RP0, RP1, RP2, and RP3. PCIe RP Cluster 1 has its PPBIFCTL located at offset C0h in Configuration Space at B0,D14,F0 and is aliased in Configuration Space at F0 for Devices 15, 16, and 17. It represents the lane arrangement of RP4, RP5, RP6, and RP7.

### 10.5.3 Configuring PCIe Lane to x1 and SATA Ports

The FIA is flexible in its lane assignments and efficient. A PCIe Root Port that results as a x2 from the bifurcation setting, can give up its higher-numbered lane so that the HSIO lane can be configured to a SATA port. Soft Straps settings allow this PCIe x1 Root Port configuration. Figure 10-4 shows an example of this PCIe-SATA arrangement of lanes. The RP Enable and RP Width Soft Strap values shown in the example are for Root Ports 0, 1, 2, and 3 for PCIe Cluster 0 and for Root Ports 4, 5, 6, and Cluster 1.

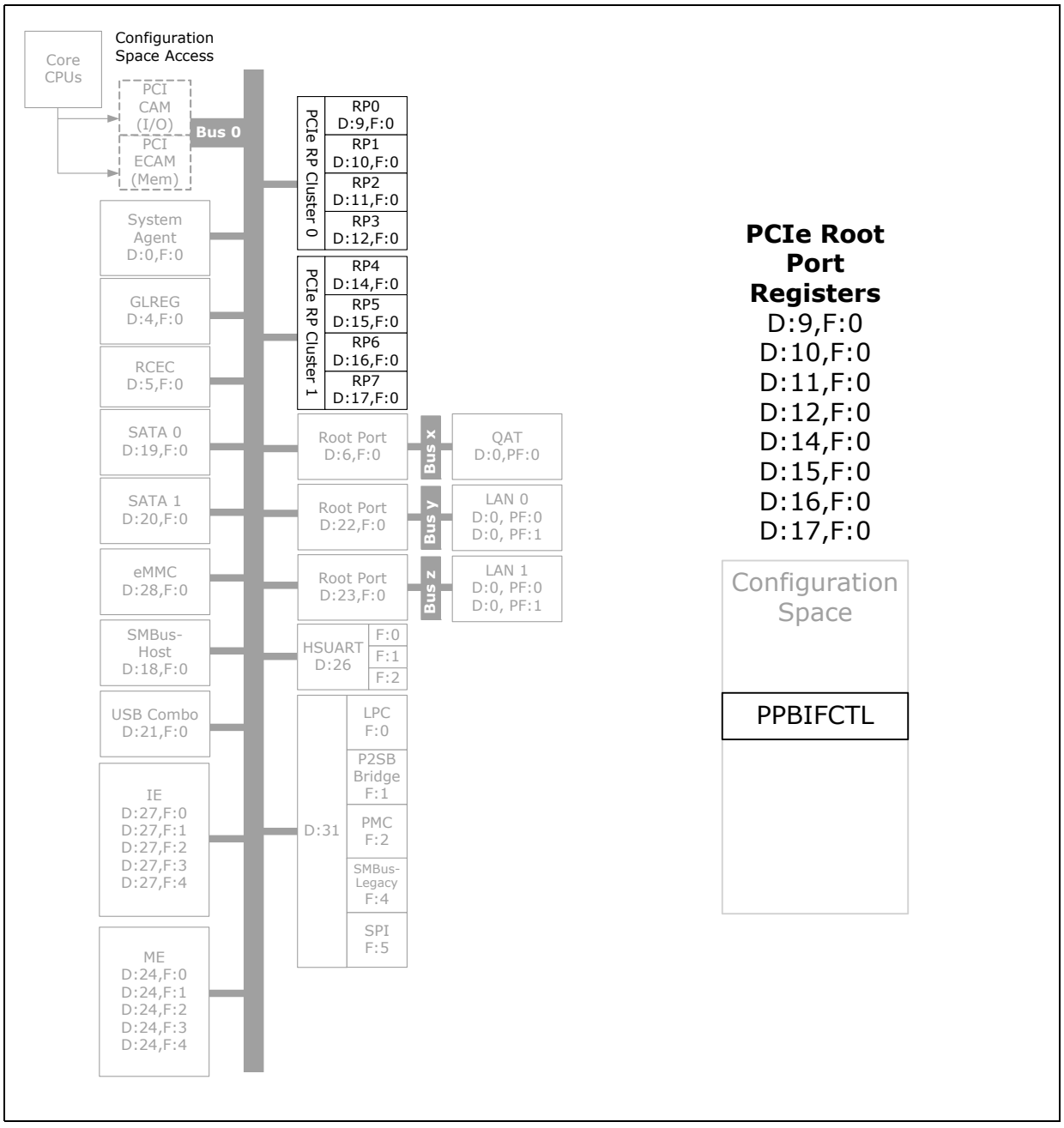
**Figure 10-4. PCIe x1 Lane and SATA Example**



## 10.6 Register Map

Figure 10-5 shows the associated registers from a system software viewpoint.

**Figure 10-5. Register Map**



### 10.6.1 PCI Configuration and Capabilities

**Table 10-3. Configuration and Capabilities Register Map**

| Offset in Configuration Space of RP0 through RP7 (hex) | Register Short Name | Description |
|---|---|---|
| C0 | PPBIFCTL | PCIe Port Bifurcation Control Register |

### 10.6.2 Sideband Registers

**Table 10-4. Sideband Registers**

| Memory-Mapped Base Address Register (BAR) | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **PCI Express Root Port Cluster 0 (RP0, RP1, RP2, RP3)** | | | |
| SBREG_BAR | B4_0010 | PPBIFCTL_PRIV | PCIe Port Bifurcation Control Register |
| **PCI Express Root Port Cluster 1 (RP4, RP5, RP6, RP7)** | | | |
| SBREG_BAR | B3_0010 | PPBIFCTL_PRIV | PCIe Port Bifurcation Control Register |

§ §

# 11 PCI Express Root Ports (RP)

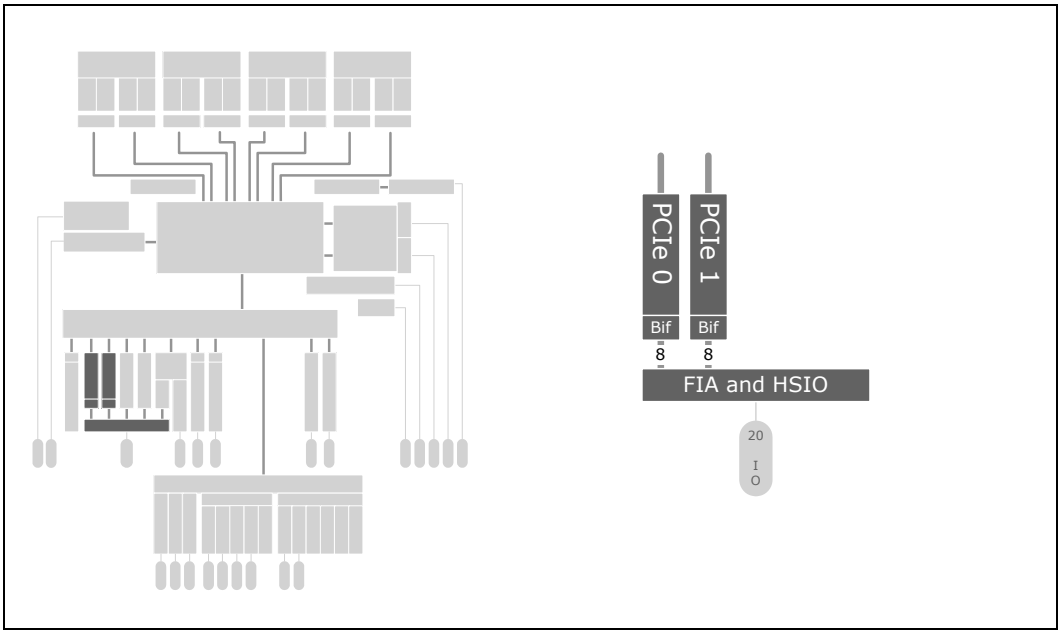
The SoC supports up to 16 PCI Express* lanes that comply with the PCI Express Base Specification Revision 3.0. The maximum data rate is 8.0 GT/s. The 16 lanes can be configured to a maximum of eight, independent PCIe* Root Ports (RP). The eight RPs are grouped in to two clusters, four RPs per cluster:

- Cluster 0 - RP0, RP1, RP2, RP3 are assigned Device Numbers 9, 10, 11, 12 decimal
- Cluster 1 - RP4, RP5, RP6, RP7 are assigned Device Numbers 14, 15, 16, 17

The number of available lanes and Root Ports depend on the product SKU.

Twelve of the 16 PCIe lanes share their SoC signal pins with the integrated SATA0 and SATA1 controllers. Each of the 12 shared lanes must be configured as either PCIe or SATA through Soft Straps and BIOS. See Chapter 10, "Flexible I/O Adapter (FIA) Overview." The Root Ports can be configured to support a diverse set of lane assignments.

**Figure 11-1. What is Covered in This Chapter**



## Table 11-1. References

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| PCI Express | 3.0 | Nov. 10, 2010 | *PCI Express Base Specification Revision 3.0* |
| PCI | 3.0 | Feb. 3, 2004 | PCI Local Bus Specification, Revision 3.0 |
| PCIe CEM | 3.0 | July 21, 2013 | *PCI Express Card Electromechanical Specification, Revision 3.0* |
| PCI Mini CEM | 2.0 | April 21, 2012 | *PCI Express Mini Card Electromechanical Specification, Revision 2.0* |
| PCI-PCI Bridge | 1.2 | June 9, 2003 | *PCI-to-PCI Bridge Architecture Specification, Revision 1.2* |
| PCI Power Management | 1.2 | March 3, 2004 | *PCI Bus Power Management Interface Specification, Revision 1.2* |

## 11.1 Signal Descriptions

The signal descriptions are shown in Table 11-2, "Signal Names and Descriptions." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names." Also, see Table 39-3, "Signals that Share the Twenty HSIO Lanes." The Direction/Type column of Table 11-2, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.
- Differential = Differential signal pair. Two pins: Positive; Negative.

**Table 11-2. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| **PCI Express RP Cluster 0 - SoC Root Ports 0, 1, 2, and 3** | | | |
| PCIE0_RP0_RP3_TX_DP[7:0]<br>PCIE0_RP0_RP3_TX_DN[7:0] | O, Differential | Yes | **Transmit (Tx):** These differential output signals are internal to the SoC. The actual signal pins are determined by the FIA configuration. |
| PCIE0_RP0_RP3_RX_DP[7:0]<br>PCIE0_RP0_RP3_RX_DN[7:0] | I, Differential | Yes | **Receive (Rx):** These differential input signals are internal to the SoC. The actual signal pins are determined by the FIA configuration. |
| PCIE_CLKREQ_N[3:0] | I,O-OD | Yes | **Reference Clock Control:** These active-low signals are used for reference clock control. See CLKREQ# signal in the PCIe Mini CEM specification. |
| **PCI Express RP Cluster 1 - SoC Root Ports 4, 5, 6, and 7** | | | |
| PCIE1_RP4_RP7_TX_DP[7:0]<br>PCIE1_RP4_RP7_TX_DN[7:0] | O, Differential | Yes | **Transmit (Tx):** These differential output signals are internal to the SoC. The actual signal pins are determined by the FIA configuration. |
| PCIE1_RP4_RP7_RX_DP[7:0]<br>PCIE1_RP4_RP7_RX_DN[7:0] | I, Differential | Yes | **Receive (Rx):** These differential input signals are internal to the SoC. The actual signal pins are determined by the FIA configuration. |
| PCIE_CLKREQ_N[7:4] | I,O-OD | Yes | **Reference Clock Control:** These active-low signals are used for reference clock control. See CLKREQ# signal in the PCIe Mini CEM specification. |
| **Common to All Root Ports** | | | |
| CLK_OUT_DP[4:0]<br>CLK_OUT_DN[4:0] | O, Differential | No | **Differential Reference Clock:** These differential output signals can be used to provide a PCIe interface clock to external PCIe devices. These five sets of output signals enable the platform board design to have five PCIe devices without requiring a clock-buffer component.<br>***Note:*** While there are eight PCIE_CLKREQ_N signal pins, there are only five Differential Reference Clock pins provided. Clock buffering may be needed on some platform board designs. |

## 11.2 Feature List

The following table lists PCI Express Root Port (RP) supported features:

**Table 11-3. PCI Express Features (Sheet 1 of 2)**

| Feature | Root Port (RP) |
|---|---|
| Alternate Routing-ID Interpretation (ARI) | Supported |
| Link Speed | Gen3[1] (8 GT/s), Gen2 (5 GT/s), Gen1 (2.5 GT/s) |
| Port Configuration | 2 RP of x8 each (16 lanes) |
| Maximum Payload Size | 256 bytes (default). 128 bytes also supported. |
| Lane reversal | Inferred during Link training rather than strapped |
| PME logging with interrupt for WAKE support | Supported |
| NP request Splitting | Supported on 64/128/256 bytes boundary |
| Posted Transaction Splitting | Supported on 64 byte boundary |
| PLL Shutdown/Standby | Supported |
| PCI Express Card Hot-Plug*<br>In-band Hot Plug | Supported |
| Full Hot-Plug (PCIe Cards + SHPC) | Supported |
| Routing of AtomicOps | Not supported |
| Revise ASPM | Supported |
| Latency Tolerance Reporting (LTR) | Not Supported |
| Project Specific Messages | Supported |
| Dynamic Link Width | Supported (as Slave, Controlled by CSR when port is Master) |
| Completion Combining | Supported |
| Windows Hardware Error Architecture (WHEA) | Supported |
| Completion Time Out (CTO) | Supported |
| End-to-End CRC (ECRC) | Supported |
| Stop and Scream (S&S) | Supported |
| Enhanced Downstream Port Containment (eDPC) | Not supported |
| Enhanced Live Error Recovery (eLER) | Not supported |
| Single Root I/O virtualization (SR-IOV) | Supported |
| Swizzle In-bound INTx into INTA | Supported |
| Separate Refclk Independent SSC Architecture (SRIS) | Supported |
| Continuous Time Linear Equalizers (CTLE) | Supported |
| Initial FoM Search Algorithm/Modified F0M Search Algorithm for WMP | Supported |
| Squelch Dynamic Enable | Supported |
| Resizable BAR | Not Supported |
| Optimized Buffer Flush/Fill Mechanism with WAKE# (OBFF) | Not Supported |
| Thermal Throttle | Not Supported |
| SATA Over PCIe | Not Supported |
| Programmable Device Idle Duration | Not Supported |

## Table 11-3. PCI Express Features (Sheet 2 of 2)

| Feature | Root Port (RP) |
|---|---|
| Address Translation Services (ATS) | Not Supported |
| TLP Processing Hints (TLP) | Not Supported |
| Dynamic Power Allocation (DPA) Capability | Not Supported |

1. SKU dependent.

## 11.3 Architectural Overview

The SoC supports two sets of PCIe* Root Port (RP) clusters. Each cluster consists of four Root Ports. Each cluster can be configured to provide up to eight lanes. RP[3:0] are assigned Device Numbers 12, 11, 10, and 9 while RP[7:4] are assigned 17, 16, 15, and 14. Each can operate at speeds up to 8.0 GT/s.

The internal bus of these root ports is Bus 0 and consists of secondary PCIe buses of specific link width. BIOS and/or OS can assign any logical bus number to the secondary PCIe ports.

Each of the eight RP controllers has three regions in Configuration Space. The contents of these regions and their offset values are:

1. Standard PCI Header:
   — Type 1 = PCI-to-PCI bridge
2. PCI Device Dependent Region:
   — PCI Express Capability
   — PCI Power Management Capability
   — SVID/SID Capability
   — Message Signaled Interrupts (MSI) Capability
   — Various implementation-specific and Intel-reserved registers
3. PCI Express Extended Configuration Space:
   — Advanced Error Reporting (AER) Capability
   — Access Control Services (ACS) Capability
   — Windows* Hardware Error Architecture (WHEA) Capability
   — Latency Tolerance Reporting (LTR) Capability
   — L1 Power Management Substates Capability

*Note:* The Multicast and Secondary PCI Express Extended Capabilities are not supported.

The PCIe Architecture also includes:

- PCIe Transaction Layer Registers
  — Performance Monitor
  — Local Event Triggers
  — Error Injection
  — Debug Registers
  — Test and High-Volume Manufacturing Registers
- PCIe Link Layer Registers
- PCIe Physical (PHY) Layer Registers

The vendor-specific extended capabilities in Configuration Space are primarily for Intel debug and testing purposes.

*Note:* The SoC does not support peer-to-peer transactions between RPs.

### 11.3.1 PCI Bridge Subsystem Identification Capability

The PCI Bridge Subsystem Vendor ID (Intel® Corporation) and additional identification information are read by the software.

### 11.3.2 Message Signaled Interrupt (MSI) Capability

Supported by the root ports:

- Per-vector masking capable (PVM)

Not Supported by the root ports:

- Address 64-Bit Capable - The root ports are not capable of generating a 64-bit message address
- Multiple Message Capable - No, only one message is supported

Supported by the root ports when enabled by the software [Default]:

- MSI Enable (MSIE) - When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. Default is 0 (not set)
- Trigger Mode (TM) - Either Edge-Triggered or Level-Triggered. Default is Edge-Triggered
- Level- or edge-triggered messages are always treated as assert messages. For level-triggered interrupts, the Level bit reflects the interrupt input state if TM (above) specifies the level-triggered mode as follows:
  - 0: Deassert messages
  - 1: Assert messages

### 11.3.3 Advanced Error Reporting (AER) Capability

PCI Express* defines two error reporting paradigms:

- Baseline capability
- Advanced Error Reporting (AER) capability

The baseline error reporting capabilities are required of all PCI Express devices and define the minimum error reporting requirements. The root ports provide the optional Advanced Error Reporting Capability which is defined for more robust error reporting and is implemented with a specific PCI Express Capability structure. PCIe* baseline error handling does not support severity programming.

The Advanced Error Reporting Capability provides each of the root ports the Uncorrectable Error Severity register which allows each uncorrectable error to be programmed to fatal or non-fatal. Uncorrectable errors are not recoverable using defined PCI Express mechanisms. However, the SoC considers a particular error fatal to a link or device or possibly considers that error non-fatal. The Uncorrectable Error Severity register (ERRUNCSEV) default value is re-programmed if the device driver or platform software requires more robust error handling.

## 11.3.4 PCIe Extended Capability and Other Supported Technology

### 11.3.4.1 Access Control Services (ACS) Capability

ACS prevents various forms of silent data corruption by preventing PCI Express Requests from being incorrectly routed to a peer Endpoint below a switch. ACS is also used to validate that every request transaction between two downstream components is allowed. Also, ACS allows some robustness checks by checking for the ReqID from a function to be a valid ReqID at a coarse granularity. When ACS is enabled, the PCIe Root Port does loopback memory transactions (reads and writes) to the same port if the address map check matched.

### 11.3.4.2 Windows Hardware Error Architecture (WHEA)

WHEA is an error-injection methodology that allows the software stack to test how it handles an error. WHEA is a Windows* hardware-enabling specification that requests one correctable and one uncorrectable error to be injected in the SoC.

WHEA is implemented in the PCIe Root Ports controllers as a vendor-specific error-injection capability structure, defined for software recognition in Configuration Space. The Intel-defined Vendor Specific ID is 3h.

It is a programmatic way to inject a correctable and an uncorrectable error. Two bits are provided to allow software to inject correctable and uncorrectable errors. Errors are injected when software sets one of the bits. Software must clear the error injection bit before setting again to generate another error. The bit will not be cleared by the hardware.

This will cause normal error logging and interrupt trigger. The error handler code is expected to be invoked. Error injection software should not preclude the invocation of the error handler. The appropriate status registers are set so that the operating system handler can go through its flows.

As a security feature, a write-once bit is provided to disable the error injection mechanism. BIOS sets this bit in every boot to prevent a denial-of-service (DOS) attack. BIOS keeps this disable bit clear only when error injection is being tested.

The error-injection mechanism does not require dependency on any external component.

The selected correctable error is the Receiver Error. The uncorrectable error is the Completion Timeout Error. Neither of these have header logs associated with them.

### 11.3.4.3 Dynamic Link Width

The capability allows either end of a link to initiate a dynamic link width change by going through Recovery and Configuration back to L0. This can be used by devices to adjust the link width based on dynamically changing bandwidth requirements. The UPCFGCSR register, located at offset AB8h in the Root Port's Configuration Space, allows the device to advertise up configure capability in the TS2 training set, so the other end can dynamically adjust the link width as needed.

### 11.3.4.4 Alternative Routing-ID Interpretation (ARI)

ARI enables next generation I/O implementations to support an increased number of concurrent users of an Endpoint, while providing the same level of isolation and controls found in existing implementations. While ARI obviously benefits the virtualized operating environments where each Requester ID can be uniquely assigned to a guest OS, ARI also benefits non-virtualized environments where (due to increased process improvements) a large number of I/O Functions can be integrated into a single Endpoint.

Routing IDs, Requester IDs, and Completer IDs are 16-bit identifiers traditionally composed of three fields:

- an 8-bit Bus Number
- a 5-bit Device Number
- a 3-bit Function Number

With ARI, the 16-bit field is interpreted as two fields instead of three:

- an 8-bit Bus Number
- an 8-bit Function Number

The Device Number field is eliminated.

This new interpretation enables an ARI Device to support up to 256 Functions. ARI is controlled by a new set of capability and control register bits defined in the PCIe registers.

## 11.4 Physical Layer

### 11.4.1 PCI Express Speed Support

The PCI Express (PCIe) physical layer implements high-speed, low-voltage differential signaling that is described in detail in the PCI Local Bus Specification, Revision 3.0. The integrated root ports support 2.5 GT/s, 5 GT/s and 8 GT/s (SKU dependent) PCI Express speeds. The Root Port controllers negotiate the speed using the in-band signaling mechanism, Link Initialization and Training, of the specification.

The PCI Express Root Ports use 8b/10b encoding when the data rate is 2.5 GT/s or 5 GT/s. For data rates greater than or equal to 8 GT/s, the encoding mechanism used is a link-wide physical layer packeting along with a per-lane encoding scheme that uses 2-bits of sync header with a payload of 128 bits.

### 11.4.2 Separate Refclk with Independent SSC Architecture (SRIS)

SRIS can be utilized to enable a PCI Express application involving an inexpensive cabling solution such as for PCIe-attached Solid-State Storage Devices (SSD) with a new cabling form factor that supports non-common clock mode with Spread-Spectrum Clocking (SSC) architecture. The reference clock is not needed to be delivered through cable. However, the new clock mode requires both sides of a link to tolerate a much higher ppm tolerance of 5600 ppm compared to the PCIe Base Specification defined as 600 ppm. For more details, please refer to PCI-SIG Engineering Change Notice (ECN) "Separate Refclk Independent SSC Architecture (SRIS)."

- The SoC has Soft Straps to enable SRIS on each of the eight Root Ports. See Table 4-3, "Flash Descriptor Soft Straps."
- The SoC Root Ports only support Separate Refclk with no SSC (SRNS) and do not support SRIS for 8-GT/s speeds. Both SRIS and SRNS are supported for PCIe 2.5-GT/s and 5-GT/s speeds.

### 11.4.3 Form Factor Support

The PCIe controllers support card-edge-connector and Server I/O Module (SIOM) form-factors. Form-factor-specific differences that exist for Hot-Plug* and power management are captured in their individual sections elsewhere in this chapter.

The root ports have enough buffering to provide full performance using up to a 20-inch trace of FR4 with two connectors. They do not provide any additional buffering for cable/repeater latencies and are not able to achieve full bandwidth on PCI Express using these topologies. But functionally, they are able to support cables.

### 11.4.4 Configuration of the PCI Express Ports and Link Widths

While there are up to two, eight-lane (Maximum Link Width = 8) PCI Express Root Port clusters, each can be configured to provide one, two, three or four Root Ports. Each of the two controllers has a PCI Express Port Bifurcation Control (PPBIFCTL_PRIV) register that determines the number of Root Port Links per controller and the number of Lanes per Link. The SoC also provides flexibility in the number of High-Speed I/O interface signals that can be made available on the SoC. Details of the PCI Express Root Port and Lane-configuration possibilities are explained in Section 10.5.2, "PCI Express Lane Configuration" of the Flexible I/O Adapter (FIA) chapter of this document.

The PPD0.ILINKTRN0 bit, located in Host Configuration Space for each of the eight possible Root Ports, controls the PCIe Express port Link Training for the associated port. A value of '1' initiates Link Training. After writing this bit to a '1', software can poll the Data Link Layer Link Active (DLLLA) bit in the LINKSTS register, which is also located in Host Configuration Space for each Root Port. Polling this bit allows software to determine if a port is up and running. Root ports do not automatically initiate Link Training after reset unless the default value of '0' of the ILINKTRN0 bit is changed to a '1'. A write of '0' to a default register-setting has no effect.

A Capability Lock (CL) bit is provided for each PPD0 register and each PPBIFCTL_PRIV register. A write of '1' locks-down the associated register for security purposes. Refer to the register-description chapters of this document for details.

There is a dedicated Soft Strap for port enable on a per-port basis to enable/disable the port. When certain PCI Express lanes within a port are disabled, via Soft Straps or through the Host Configuration-Space Registers, then the Link-Training and Status State Machine (LTSSM) associated with those lanes must be put in the "disabled" state (which disables receiver detect sequence on those lanes). In addition, strap of slot width, per port basis, can be utilized for constraining the port operation at a particular width. This is instant and the behavior is an undefined post, allowing the link to train. The SoC Soft Straps are described in Chapter 4, "Strapping and Configuration."

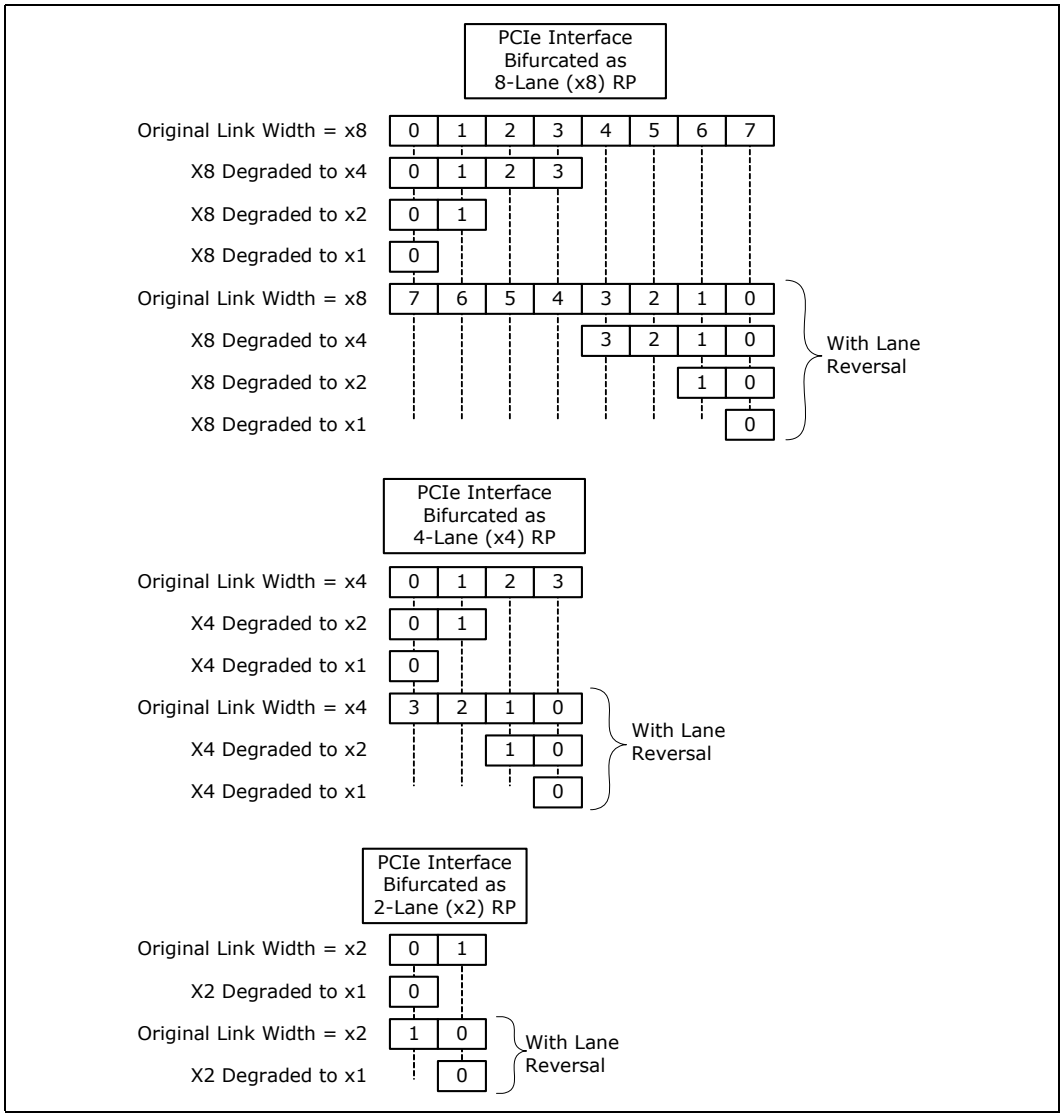
### 11.4.5 Degraded Mode

Degraded-Mode Link Widths a PCI Express negotiation behavior supported by the SoC and is automatically attempted every time the PCI Express link is trained. To maximize performance, the SoC first attempts Degraded Mode of a higher Link Width before trying any of the lower-Link-Width Degraded Modes. The x1 is the smallest Link Width supported by the SoC in the PCIe Degraded Mode operation. Link-Width degradation is always be anchored to a slot edge.

Lane-degradation remapping occurs in the Physical Layer. The Link Layer and Transaction Layer are transparent to the Link-Width change.

See Figure 11-2, "PCIe Bifurcation and Physical Ports."

**Figure 11-2. PCIe Bifurcation and Physical Ports**



### 11.4.5.1 Dynamic Lane Reversal

Lane Reversal is a Link negotiation behavior that creates a condition where if a Link fails to train, the Link is then forced to train again with its Lanes re-ordered in the opposite order. For an 8x situation with a Link wired as Lanes 0 through 7, Lane Reversal tries the training again with the Link physically wired as Lanes 7 through 0. See Figure 11-2, "PCIe Bifurcation and Physical Ports."

If a Link encounters a "bad Lane" during its initial training, the Link then chooses a "subset" of Lanes. It does this either on the reversed order, anchored to physical Lane N-1, or on the non-reversed order, anchored to physical Lane 0.

If the device connected to the Root Port does not support Lane Reversal, Link training may fail (no possible Lane-0 match).

This Lane Reversal negotiation is only enabled when the Link falls back to the Detect State from the Configuration.Lanenum.Wait or Configuration.Lanenum.Accept state prior to when the Physical Layer reports Physical LinkUp = 1b. Once the link reaches the Configuration.Complete substate, the Lane Reversal negotiation is turned off.

### 11.4.5.2 Lane Polarity Inversion

Lane Polarity Inversion is a Link negotiation behavior that is supported by the SoC. Polarity inversion allows the board designer to connect the D+ and D- lines to the Rx- and Rx+ pins of the chip respectively.

## 11.5 Transaction Layer

PCI Local Bus Specification, Revision 3.0 defines a field in the header called the Transaction Descriptor. This descriptor comprises three sub-fields:

- Transaction ID
- Attributes
- Traffic Class

### 11.5.1 Transaction ID

The Transaction ID uniquely identifies every transaction outstanding on the PCI Express interface. The Transaction ID comprises four sub-fields described in Table 11-4, "PCI Express Transaction ID Handling."

Table 11-4, "PCI Express Transaction ID Handling" provides details on how this field in the PCI Express header is populated by the SoC on the requests it issues on PCI Express (requests issued by core) and the completions it sends back on PCI Express.

*Note:* The SoC does not support Peer-to-Peer transactions between PCIe Root Ports.

**Table 11-4. PCI Express Transaction ID Handling**

| Field | Definition | SoC as Requester | SoC as Completer |
|---|---|---|---|
| Bus Number | Specifies the Bus Number on which the Requester resides. | The SoC fills this field with its internal Bus Number that the PCI Express cluster resides on. | This field from the request is preserved and copied into the completion. |
| Device Number | Specifies the Device Number of the Requester. | The SoC fills this field with the Device Number that the root port owns. | |
| Function Number | Specifies the Function Number of the Requester. | The SoC fills this field with the Function Number that the root port owns. | |
| Tag | Identifies a unique identifier for every transaction that requires a Completion. Since the PCI Express ordering rules allow read requests to pass other read requests, this field reorders separate completions if they return from the target out-of-order. | • NP Tx: The SoC fills this field with a value such that every pending request carries a unique Tag.<br>• NP Tag[7:5]=Note that bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, the SoC always zeros out 7:5.<br>• NP Tag[4:0]=Any algorithm that guarantees uniqueness across all pending NP requests from the port.<br>• P Tx: No uniqueness guaranteed. Forward from source port to destination for p2p posteds. | |

## 11.5.2 Attributes

The Attributes field is used to provide additional information that allows modification of the default-handling of Transactions. Attributes are hints that allow for optimizations in the handling of traffic. PCI Express supports two attributes:

**Table 11-5. PCI Express Attribute Handling**

| Attribute | Definition | PCIe SIP as Requester | PCIe SIP as Completer |
|---|---|---|---|
| Relaxed Ordering | Allows the system to relax some of the standard PCI ordering rules. | This bit is not applicable and set to zero for transactions that the PCIe SIP generates on PCIe on behalf of a request. | The PCIe SIP preserves this field from the request and copies it into the completion. |
| Snoop Not Respond | This attribute is set when an I/O device controls coherency through software mechanisms.<br>This attribute is an optimization design to preserve processor snoop bandwidth | | |

For relaxed ordering, PCIe ports implements Downstream CPL relaxed ordering. For the cases where the No-Snoop attribute is set in the inbound TLP, it should be passed to the upstream block or root port and the request should be treated as non-coherent in the UnCore.

The PCI express ports support the “No Snoop Attribute” but there is no isochronous support.

### 11.5.3 Traffic Class

The SoC does not support any PCI Express Virtual Channels (VC) except for the default channel VC0. There is no Traffic Class (TC)-to-VC mapping capability structure. However, as per the PCI Express Base Specification, a completer must accept requests with a TC label other than TC0, and must preserve the TC label.

The SoC accepts inbound transactions with zero/non-zero TC, but it treats the non-zero TC as mapped to VC0 for inbound requests destined for memory. For completions heading upstream, the Root Port copies the TC value received into the completion packet. If the SoC receives a non-zero TC for any I/O or Configuration transaction, it is treated as a malformed TLP and the error-logged appropriately.

### 11.5.4 Completer ID

The CompleterID field is used in PCI Express completion packets to identify the completer of the transaction. The CompleterID comprises three sub-fields which are populated on a completion sent back on PCI Express, as described in Table below.

**Table 11-6. PCI Express CompleterID Handling**

| Field | Definition | PCIe RP as Completer |
|---|---|---|
| Bus Number | Specifies the bus number om which the completer resides on. | PCIe RP fills this field with its internal Bus Number on which the port resides. |
| Device Number | Specifies the device number of the completer. | PCIe RP fills this field with its Internal Device Number. |
| Function Number | Specifies the function number of the completer. | PCIe RP fills this field with its Internal Function Number. |

## 11.5.5 Transaction Ordering

Table 11-7 lists the combined set of ordering rules in the PCIe inbound paths of the root port.

**Table 11-7. Transaction Ordering on PCIe Block**

| Row Pass Column | Message or Posted Wr (PR) | Rd or Configure Write or I/O Req (NPR) | Rd Completion | I/O or Configure Write Completion |
|---|---|---|---|---|
| Message or Posted Wr (PR) | No | Yes | Yes | Yes |
| Rd or Configure Write or I/O Req (NPR) | No | No | Yes[1] | Yes |
| Rd Completion | No | Yes | No | No |
| I/O or Configure Write Completion | No | Yes | No | No |

1. When a completion is stalled, a subsequent read can be allowed to pass it.

## 11.5.6 PCI Express Port Arbitration

The SoC provides a normal, weighted (based on port width), round-robin method of arbitration amongst the eligible sources.

## 11.5.7 Request Splitting

The Transaction Layer Packet (TLP) Maximum Payload size is defined as 256 bytes for the PCI Express* Root Ports.

Each SoC Root Port splits Non-Posted Requests greater than 256 bytes at the 256-byte address boundary. This maximum-byte size of 256 is without regard to the target address. The expectation is, that for better performance, the target agent will provide the maximum payload-size completions. The Root Port implements completion tracking to assure that multiple completions for a greater-than 256-byte read are sent to the requester in addressed order.

Posted Requests in the Root Port have a length of up to the defined Maximum Payload Size of 256 bytes and are handled as normal TLPs.

### 11.5.8 Read Prefetching Policies

The Root Ports do not perform any outbound-read prefetching. They also do not perform any autonomous, speculative Prefetch on their inbound interfaces.

### 11.5.9 Read Completion Combining

The PCI Local Bus Specification, Revision 3.0 allows a single request to be satisfied with multiple "sub-completions" as long as the sub-completions return in linearly-increasing address order. Therefore, since the Root Ports must split requests into cacheline quantities before issuing them to the coherent interface, the Root Ports often complete a large request in cacheline-sized sub-completions.

As a performance optimization, the Root Port implements an opportunistic read-completion combining algorithm. When the downstream PCI Express* interface is busy (for example, with another transaction) and multiple cachelines have returned before completion on PCI Express is possible, the PCI Express interface combines the cacheline sub-completions into larger quantities up to defined Maximum Payload size.

### 11.5.10 Write Combining

The Root Ports do not support any outbound write combining.

Potentially, inbound memory writes to the system DDR SDRAM could be combining in the Root Port Write of the coherent interface.

## 11.6 Interrupt Support

The PCIe Root Port supports both Message Signaled Interrupt (MSI) and in-band legacy interrupt mechanisms. The PCIe Root Port can generate an in-band legacy interrupt request on PCI Express for boot devices and for systems that do not support MSI.

### 11.6.1 Legacy Interrupt Sharing

PCI Express provides interrupt messages that emulate the legacy wired mechanism.

This feature allows I/O devices to signal PCI-style interrupts using a pair of ASSERT and DEASSERT messages. This message pairing preserves the level-sensitive semantics of the PCI interrupts on PCI Express.

The four virtual wire interrupts (INTA–INTD) correspond to the four interrupt wires defined in the PCI Local Bus Specification, Revision 3.0. The switch routes its interrupts messages from the downstream ports to the upstream ports interrupts as shown in Table below.

**Table 11-8. Bridge Mapping for Interrupt Virtual Wires**

| Device # | Interrupt Message on Port | Interrupt Message on the Upstream Port |
|---|---|---|
| 0, 4, 8, 12, 16, 20, 24, 28 | INTA# | INTA# |
| | INTB# | INTB# |
| | INTC# | INTC# |
| | INTD# | INTD# |
| 1, 5, 9, 13, 17, 21, 25, 29 | INTA# | INTB# |
| | INTB# | INTC# |
| | INTC# | INTD# |
| | INTD# | INTA# |
| 2, 6, 10, 14, 18, 22, 26, 30 | INTA# | INTC# |
| | INTB# | INTD# |
| | INTC# | INTA# |
| | INTD# | INTB# |
| 3, 7, 11, 15, 19, 23, 27, 31 | INTA# | INTD# |
| | INTB# | INTA# |
| | INTC# | INTB# |
| | INTD# | INTC# |

The switch tracks the INTx messages from each port independently and presents a collapsed INTx message on the upstream port. For example, each INTx message from the downstream ports are ORed together and presented upstream as a single INTx message. The switch uses its upstream port primary bus number and device number in the Requester ID field for the PCI Express INTx messages. As stated in the PCI Local Bus Specification, Revision 3.0, the function number is reserved for interrupt messages and is always 0.

## 11.7 Power Management

### 11.7.1 Hardware Controlled Active Power Management

PCI Express Base Specification defines a HW-initiated power management of the PCIe link called the active state power management (ASPM). Under hardware control, the link can be put into a low-power L0s state or an even lower power L1 link state. Latency to recover from L0s back to L0 (active link state) is smaller than that of recovery from L1 to L0. The PCIe Root Port supports both ASPM link states L0s and L1. ASPM is totally traffic dependent and is not initiated by software. But the software could either enable or disable the active state management via the PCIe Express capability structure.

### 11.7.2 PCI-PM 1.2 Support

The PCI Express Root Port supports PCI Express link states required to implement PCI_PM 1.2 compatible device states (D0, $D3_{HOT}$, $D3_{COLD}$). These states correspond to PCI Express link states L0, L1, and L3. When all the root ports and the IOxAPIC are programmed to $D3_{HOT}$ state, the upstream link of the switch automatically transitions to the link state L1. The switch also supports the PME_Turn_Off/PME_TO_Ack handshake protocol to enter the $D3_{COLD}$/L3 device/link states.

Each function (upstream port, downstream ports, and IOxAPIC) behaves as follows when in the $D3_{HOT}$ state:

- The function responds to Configuration cycles on UpStream PCIe ports
- The function does not respond to Memory Cycles on UpStream PCIe ports except for completions
- The function does not respond to I/O cycles on UpStream PCIe ports except for completions
- The function does not initiate UpStream PCIe port transactions
- The function does not reset its registers when programmed from D0 to $D3_{HOT}$

*Note:* When a downstream port is in the $D3_{HOT}$ state, a Hot-Plug event on that port must generate a PME event since the port is unable to generate an interrupt. Software needs to setup the Hot-Plug to generate a wake event instead of interrupts when the port is in $D3_{HOT}$ state.

### 11.7.3 Power Management Event Signaling

PCIe Root Port supports Active State Power Management (ASPM) transitions into L0s and L1 state. L0p state is not supported by the SoC.

Also, the PCIe Root Port supports the D0 and D3hot power management states per PCI Express port and also supports a wake event from these states on a PCI Express Hot-Plug event. In D3hot, the root port will master abort all configuration transactions targeting the PCI Express link.

PME signaling in PCI-Express accomplishes two distinct functions:

- Firstly, it provides a signaling mechanism for devices requiring service to propagate a wake-up request to the power management controller.
- Secondly, it provides a messaging mechanism for devices requesting a power state change to pass their unique location within the PCI Express hierarchy to the power management controller.

The combination of these two functions provides great flexibility and controllability for the power manager.

### 11.7.4 Beacon and WAKE# Signaling

At the link level, the PCI Local Bus Specification, Revision 3.0 describes two optional mechanisms used by components to request the reapplication of main power (transition to the fully-operative L0 state) when in the low-power L2 Link state:

- Beacon (using in-band signaling):
  - — Not supported by the Root Ports.
- WAKE# (using sideband signaling):
  - — Supported by the Root Ports through the signal pin WAKE_B.

### 11.7.5 Non-D0 State PM Handling

The Non-D0 and Bus Master Enable (BME) state requirements are as follows:

#### 11.7.5.1 Non-D0 Behavior for Transmit (Tx) Transactions

- Memory and I/O requests are treated as Unsupported Request (UR) completions
- Memory or I/O requests that require response get a UR response (internally generated)
- Type0 Cfg requests proceed normally for Downstream Port (of an Upstream Component)
- Type0 Cfg requests are treated as URs (completion w/ UR response) for Upstream Port (of an Downstream Component)
- Type1 Cfg are treated as URs (completion w/ UR response)
- Messages and completions are unaffected by device states

#### 11.7.5.2 Non-D0 Behavior for Receive (Rx) Transactions

- Memory and I/O requests are treated as URs
- Memory or I/O requests that require response get an UR response (internally generated)
- Type0 Cfg request is treated as URs (completion w/ UR response) for Downstream Port (of an Upstream Component)
- Type0 Cfg request proceeds normally for Upstream Port (of an Downstream Component)
- Type1 Cfg is treated as URs (completion w/ UR response)
- Messages and completions are unaffected by device states

When the Upstream port is programmed to the non-D0 state, it should initiate its Upstream PCIe link to the transition to PCI-PM L1 state when all conditions are met.

The transition of a PCIe of a Virtual Bridge (such as a root port) from D3hot state to D0 because of a powerstate command will not cause an internal soft reset and this is indicated by setting the No_Soft_Reset bit3 (RO) of PMCSR register.

### 11.7.6 BME Clear State Handling

In BME (=1) set state, all transactions are treated as normal.

#### 11.7.6.1 BME (=0) Clear State Behavior for Transmit (Tx) Transactions

- Downstream Port (of an Upstream Component)
  - All transmit transactions are treated as normal and behavior is unaffected by BME bit state
- Upstream Port (Downstream Component)
  - Memory, I/O requests are treated as URs
  - Configuration, Messages and completions are treated as normal and behavior is unaffected by BME bit state
  - All requests that require response get an UR response (internally generated)

#### 11.7.6.2 BME (=0) Clear State Behavior for Receive (Rx) Transactions

- Downstream Port (of an Upstream Component)
  - Memory, I/O requests are treated as URs
  - Configuration, Messages and completions are treated as normal and behavior is unaffected by BME bit state
  - All requests that require response get an UR response (internally generated)
- Upstream Port (of an Downstream Component)
  - All transmit transactions are treated as normal and behavior is unaffected by BME bit state

#### 11.7.6.3 Internal MSI/Error Messages During Non-D0/BME (clear)

- MSIs generated internally are blocked when in non-D0 and BME (clear).
- MSIs generated internally in non-D0 with BME = 1 (for example, a completion timeout or an MSI due to PME event) should be allowed to propagate but after the port is reinitialized back to D0. There should be no loss of interrupts.
- MSIs received at the root port or in the switch port from external sources should be blocked (UR) when non-D0.
  **Assumption:** PCI configuration software should place the downstream component in non-D0 first and then the upstream components. There should be no MSI writes received at the upstream component under these conditions and if any spurious MSIs do arrive, they will be master aborted.
- INTx, VLW and ERR messages should not be blocked.

#### 11.7.6.4 L1 PM Substates

PCIe supports the optional PM L1 Substates including L1.1 and L1.2,where L1.1 is hardware autonomous without Software intervention and L1.2 may require Software not to access device for longer period, which allow for dramatically lower idle power, including near complete removal of power for high speed circuits. The L1 PM Substates are applicable in both the ASPM and PCI-PM L1 Link states and the entrance and depth is constrained by aggregated PCIe LTR.

## 11.8 PCI Express RAS Features

The PCI Local Bus Specification, Revision 3.0 supports PCI Express Advanced Error Reporting (AER) capability.

### 11.8.1 End-to-End CRC (ECRC) Support

PCIe End to End CRC (ECRC) is the CRC field generated by a device and is appended to the TLP to protect the data through layers of switches and bridges.

For inbound traffics, PCIe Root Ports support ECRC checking and will strip the ECRC if no error is detected. The corresponding TLP will be dropped and logged UC if an error is detected. ECRC error severity should be set to non-fatal in the Uncorrectable Error Severity Register during initialization.

For outbound traffic, PCIe Root Ports will generate ECRC based on AER Capabilities and Control Register (AERCAPCTL), and insert ECRC to the packet prior to forwarding to the PCIe link. Refer to PCI Local Bus Specification, Revision 3.0 for more details of ECRC generation rules.

### 11.8.2 Completion Timeout

For all non-posted requests that the PCIe Root Port issues, the PCIe maintains a timer that times the max completion time for that request.

The PCIe Root Port follows the timeout mechanism as defined in the PCI Express Base Specification. The spec provides a way for the OS to select a coarse range for the timeout value. The PCIe root port chooses a final value of the timeout within each coarse range. The timeout value is programmable from 50 ms all the way up to 64s.

Refer to Chapter 6, "Reliability, Availability and Serviceability (RAS)" for details of responses returned by the PCIe Root port to various interfaces on a completion timeout event. AER-required error logging and escalation happen as well.

### 11.8.3 Data Poisoning

The PCI Express Root Ports support forwarding poisoned information between the coherent interface and the PCIe link, and vice-versa. Also, forwarding poisoned data between peer PCIe ports is supported. The PCIe has a mode where poisoned data is never sent out on PCI Express, in other words, any packet with poisoned data is dropped internally in the root port and an error escalation done.

### 11.8.4 Role-based Error Reporting

The PCIe Root Port supports the new role-based error reporting that is specified in the PCI Local Bus Specification, Revision 3.0.

When the PCIe Root Port times out or receives a UR/CA response on a request outstanding on PCI Express, it does not attempt recovery in hardware. The UR/CA received does not cause any error escalation via AER mechanism and causes error escalation only via the IO module-specific mechanism. The PCIe Root Port would forward those to the ultimate destination. Also, the PCIe Root Port would treat the completion timeout condition as a normal non-fatal error condition (and not as an advisory condition).

The PCIe Root Port also treats an unexpected completion it receives from PCI Express port as an advisory non-fatal error if the severity of it is set to non-fatal. If the severity is set to fatal, then unexpected completions are NOT treated as advisory but as fatal.

When the PCIe Root Port detects a UR/CA condition on a non-posted request it received from PCIe link, it treats the error as advisory error if the severity of UR/CA is set to non-fatal, and send a UR/CA completion back to the device. Note that a UR/CA condition detected on a posted request that the PCIe Root Port received from the PCIe link is treated as normal non-fatal error with the appropriate escalation/logging.

## 11.9 PCI Express Hot-Plug Surprise

PCIe Root Port Hot-Plug slot capability is supported mainly through software stacks by setting Hot-Plug related CSRs for handling Hot-Plug events such as in-band presence detect. It does not utilize any mechanical or electromechanical features such as MRL/sensor/EM interlock/ Attention button, etc.

The Hot-Plug Surprise slot capability involves a device that is inserted into or removed from a running system without prior notification to the system. It is not an orderly operation and could cause undesirable effects to the running system. The general expectation from hardware and software is to avoid Hot-Plug error event propagating into a catastrophic system event.

In general, all outstanding Non-posted request should synthesize a Master Abort to avoid completion timeout and suppress catastrophic AER storm whenever applicable. En-queued inbound traffic at the port upstream to the device must complete normally in the event of a Surprise Removal. PME_Turn_Off handshake and VDM should be terminated appropriately plus discarding all other posted & completion requests.

The Hot-Plug Surprise event is limited to the link transitions back to L0, in other words, software may need to wake up periodically during low power states accordingly. To handle the L1 inband Hot-Plug Surprise case, the link will not be allowed to train until the SLOTSTS.PDCS has been cleared by software to avoid hot-add interrupt from being merged with the hot-removal interrupt.

### 11.9.1 PCI Express Hot-Plug Interrupts and Events

#### 11.9.1.1 Presence Detect

When a module is plugged in and power is supplied, the physical layer will detect the presence of the device, and the downstream port sets SLOTSTS.PDS and SLOTSTS.PDCS. If SLOTCTL.PDCIE and SLOTCTL.HPIE are both set, the downstream port will also generate an interrupt. When a module is removed through physical layer detection, the downstream port clears SLOTSTS.PDS and sets SLOTSTS.PDCS. If SLOTCTL.PDCIE and SLOTCTL.HPIE are both set then the downstream port will also generate an interrupt. The interrupt is generated on an edge-event. For example, if SLOTSTS.PDCS is already set, a change in SLOTSTS.PDS will not generate a new interrupt. Only SLOTSTS.PDCS going from `0 to `1 will cause an interrupt to be generated.

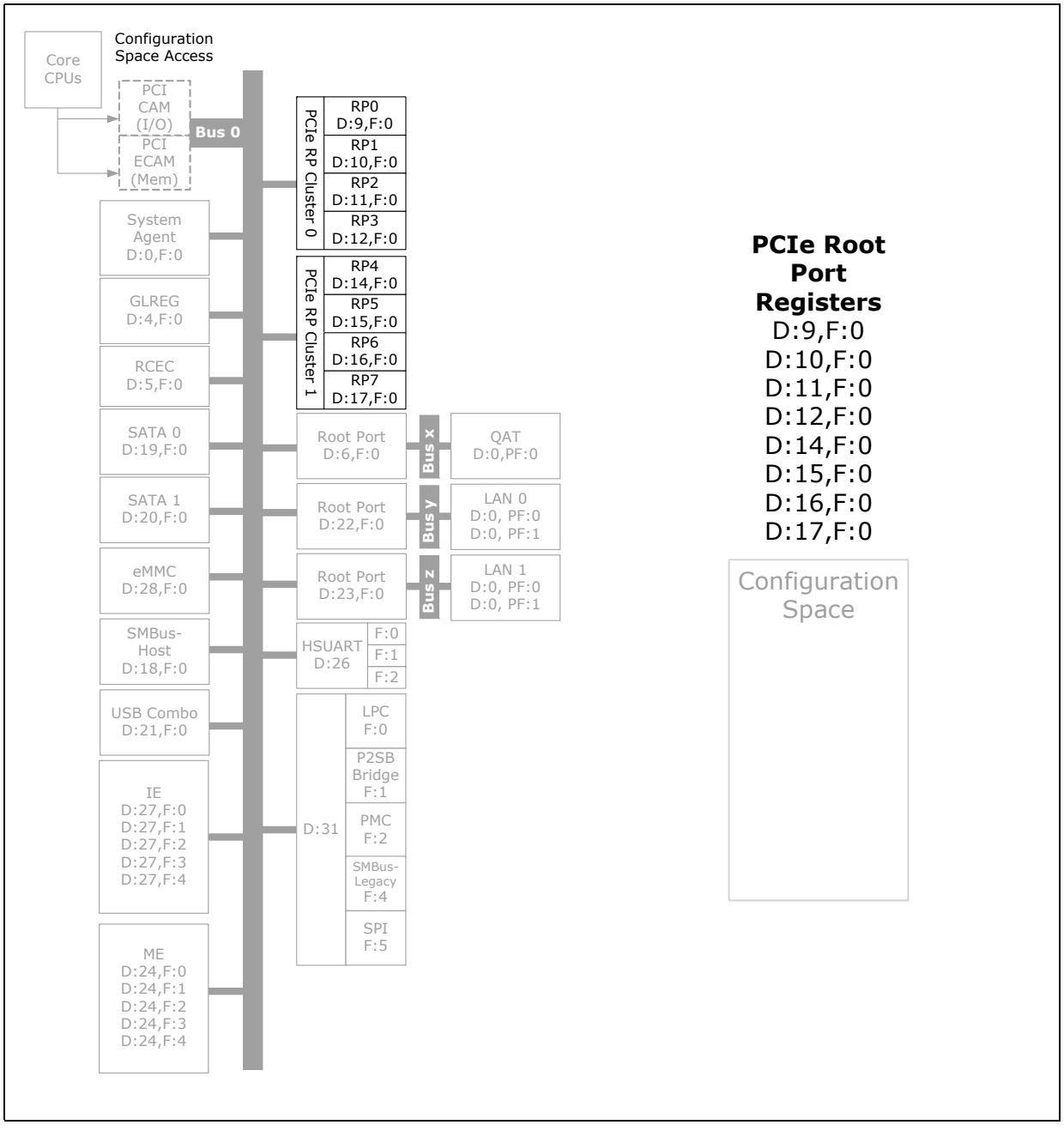
#### 11.9.1.2 SCI/SMI Generation

When a module is plugged in and power is supplied, the physical layer will detect the presence of the device, and the downstream port sets SLOTSTS.PDS and SLOTSTS.PDCS. If SLOTCTL.PDCIE and SLOTCTL.HPIE are both set, the downstream port will also generate an interrupt. Similarly, PCIe Root Port could generate SCI/SMI based on PM_PME message received from downstream PCIe devices.

## 11.10 Register Map

Figure 11-3, "Register Map" shows the associated registers from a system software viewpoint.

**Figure 11-3. Register Map**




§ §

# 12 SATA Controller

The SoC has two independent integrated SATA 3.1 host controllers (SKU dependent). Each controller supports DMA operation on up to eight ports and supports data transfer rates of up to 6 Gbps (600 MBps). Legacy data transfer rates of 3 Gbps and 1.5 Gbps are supported as well. Serial General Purpose Input Output (SGPIO) support is added in this generation and support is per the SFF 8485 specification.

*Note:* Legacy and Native IDE are not supported in this generation.

**Figure 12-1. What is Covered in This Chapter**

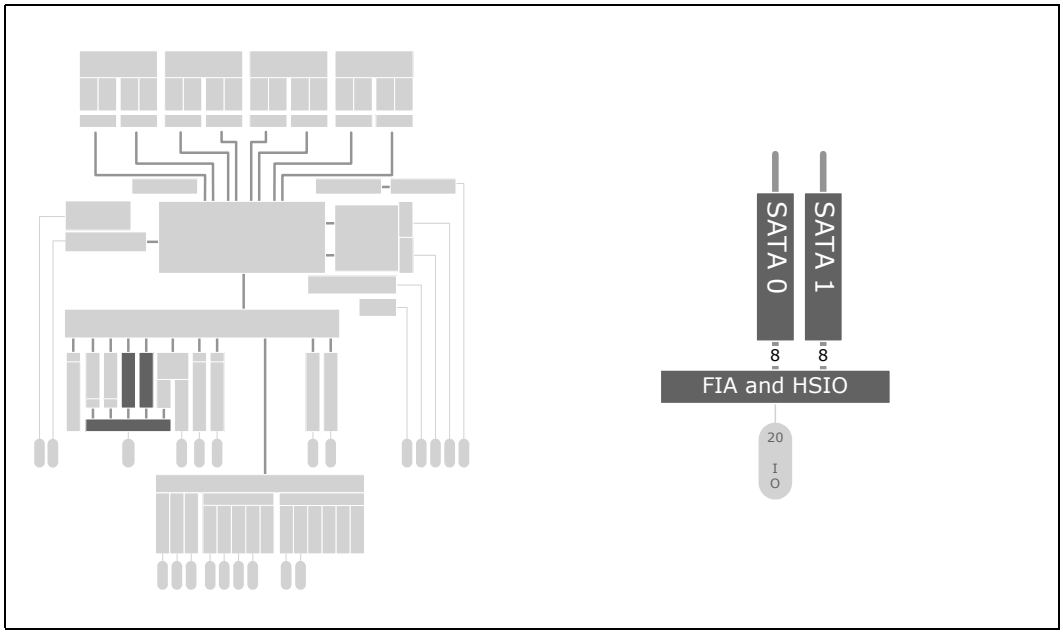



**Table 12-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| Serial ATA | 3.1 | July 2011 | *Serial ATA Specification, Revision 3.1* |
| Serial ATA | 3.0 | June 2009 | *Serial ATA Specification, Revision 3.0* |
| AHCI Specification | 1.3.1 | June 2008 | *Serial ATA Advanced Host Controller Interface (AHCI) Specification, Rev 1.3.1* |
| SFF-8485 | 0.7 | Feb. 1, 2006 | *SFF-8485 Specification for Serial GPIO (SGPIO) Bus*, Revision 0.7 |

## 12.1 Signal Descriptions

The signal descriptions are shown in Table 12-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. Also, see Table 39-3, "Signals that Share the Twenty HSIO Lanes" on page 1028. The Direction/Type column of Table 12-2 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.
- Differential = Differential signal pair. Two pins: Positive; Negative.

### Table 12-2. Signal Names and Descriptions

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| **SATA Controller 0** | | | |
| SATA0_TX_DP[7:0]<br>SATA0_TX_DN[7:0] | O, Differential | Yes | SATA0 Transmitter: These differential output signals are internal to the SoC. They are at the SATA0 controller interface with the FIA circuitry. |
| SATA0_RX_DP[7:0]<br>SATA0_RX_DN[7:0] | I, Differential | Yes | SATA0 Receiver: These differential input signals are internal to the SoC. They are at the SATA0 controller interface with the FIA circuitry. |
| SATA0_SLD | O-OD | Yes | SATA0 SGPIO Load: This is the SGPIO SLoad signal. During power-up and reset, this pin is 1 (high impedance). When using the SGPIO bus but purposely not exchanging a bit stream, the SoC sets SLoad to 0. This lets the target know that the initiator (the SoC) is still present and has not been removed. |
| SATA0_SCLK | O-OD | Yes | SATA0 SGPIO Clock: This is the 32-kHz SGPIO SClock signal. During power-up and reset, this pin is 1 (high impedance). When using the SGPIO bus but purposely not exchanging a bit stream, the SoC sets SClock to 0. |
| SATA0_SDOUT | O-OD | Yes | SATA0 SGPIO Data Out: This is the SGPIO SDataOut signal. During power-up and reset, this pin is 1 (high impedance). |
| SATA0_LED_N | O-OD | Yes | Open drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive platform LED. When active, the LED is on. When inactive, the LED is off. An external pull-up resistor is required. |
| SATA_PDETECT[1:0] | I | Yes | Mechanical Presence Detect: This input signal is called Mechanical Presence Switch in the AHCI Specification:<br>• SATA_PDETECT[0] - SATA Controller 0, Port 0 (FIA/HSIO Lane 4)<br>• SATA_PDETECT[1] - SATA Controller 0, Port 1 (FIA/HSIO Lane 5) |
| **SATA Controller 1** | | | |
| SATA1_TX_DP[7:0]<br>SATA1_TX_DN[7:0] | O, Differential | Yes | SATA1 Transmitter: These differential output signals are internal to the SoC. They are at the SATA1 controller interface with the FIA circuitry. |
| SATA1_RX_DP[7:0]<br>SATA1_RX_DN[7:0] | I, Differential | Yes | SATA1 Receiver: These differential input signals are internal to the SoC. They are at the SATA1 controller interface with the FIA circuitry. |
| SATA1_SLD | O-OD | Yes | SATA1 SGPIO Load: This is the SGPIO SLoad signal. During power-up and reset, this pin is 1 (high impedance). When using the SGPIO bus but purposely not exchanging a bit stream, the SoC sets SLoad to 0. This lets the target know that the initiator (the SoC) is still present and has not been removed. |
| SATA1_SCLK | O-OD | Yes | SATA1 SGPIO Clock: This is the 32-kHz SGPIO SClock signal. During power-up and reset, this pin is 1 (high impedance). When using the SGPIO bus but purposely not exchanging a bit stream, the SoC sets SClock to 0. |
| SATA1_SDOUT | O-OD | Yes | SATA1 SGPIO Data Out: This is the SGPIO SDataOut signal. During power-up and reset, this pin is 1 (high impedance). |
| SATA1_LED_N | O-OD | Yes | Open drain output pin driven during SATA command activity. It is to be connected to external circuitry that can provide the current to drive platform LED. When active, the LED is on. When inactive, the LED is off. An external pull-up resistor is required. |

## 12.2 Feature List

- Complies to SATA specification rev 3.1
- Complies AHCI specification rev 1.3.1
- Supports Native Command Queuing. Up to 32 commands can be queued
- Supports Staggered Spin-up
- Supports Aggressive Link Power Management
- Activity LED support via SGPIO
- Enclosure Management support
- Supports Command based Port Multiplier
- Supports Hot-Plug operation
- Supports MSI and MSI-X interrupts.

## 12.3 Architectural Overview

There are two independent SATA controllers. Both controllers support up to SATA Gen3 (SKU dependent) traffic speeds of 6 Gbps and are backward compatible to Gen2 and Gen1 speeds. Software mode is supported; this mode requires specific OS device driver support. IDE support is not available. The SoC enables SGPIO support, this is to enable and manage target device LED control. Enclosure management support is also available. SoC also has two dedicated pins for LED activity. SATA0_LED_N and SATA1_LED_N can be employed if a single drive is populated on SATA controller 0 and 1.

Each SATA controller has the following PCI capabilities:

- PCI Power Management Interface Capability
- Message Signaled Interrupts (MSI) Capability
- Extended MSI (MSI-X) Capability
- Serial ATA Capability

*Note:* The SATA controller configuration registers have default values that do not include the MSI-X Capability in the Capabilities List. BIOS must alter the Capabilities List links to include the MSI-X Capability which is located at offset D0h of each controller Configuration Space.

*Note:* If your design uses the SoC SATA then one of the following has to be met:

1. HSIO lane 16, 17, 18 or 19 must be assigned to the xHCI controller and the xHCI controller must be enabled and visible as a PCI device.

   OR

2. A PCIe Root Port Cluster must be enabled and visible as a PCI device. There is no requirement to assign any HSIO lanes to the PCIe Root Port.

## 12.4 AHCI Operation

Each controller supports up to 8 ports and each port can support up to 32 commands. A total of 256 commands can be pending at an instance per controller, a total 512 commands across controllers.

For AHCI operation details, refer to the *Serial ATA Advanced Host Controller Interface (AHCI) Specification, Rev 1.3.1*.

### 12.4.1 Staggered Spin Up Support

A staggered spin up feature enables a controller to individually spin up attached devices. This mechanism is useful to avoid having a power supply, needing to apply power to all the devices, while handling maximum current draw from all devices at the same time. In order for a system to support staggered spin up, the SATA controller, BIOS and device driver must all support staggered spin up.

Staggered spin-up is controlled by software via sequencing of COMRESET signaling. The table below summarizes the operations of SATA controller in AHCI mode. Refer to the *Serial ATA Advanced Host Controller Interface (AHCI) Specification, Rev 1.3.1* for more details on AHCI staggered spin up.

**Table 12-3. SATA Controller Operations Summary**

| PCS.PxE | CAP.SSS | PxCMD.SUD | Result |
|---|---|---|---|
| 0 | X | X | No COMRESET and no spin up |
| 1 | X | X | COMRESET and spin up |
| 0 | X | X | No COMRESET and no spin up |
| 1 | 0 | 0 | This is not a valid combination. When CAP.SSS is 0, PxCMD.SUD must be 1 |
| 1 | 0 | 1 | COMRESET and spin up |
| 1 | 1 | 0 | No COMRESET and no spin up |
| 1 | 1 | 1 | COMRESET and spin up |

### 12.4.2 Staggered Spin-up Operation in AHCI Mode

Device Driver/BIOS to perform following actions

1. Programming ABAR.
2. Configure controller in AHCI mode by setting AHCI GHC.AE bit.
3. Set AHCI CAP.SSS to indicate staggered spin up support.
4. Set PCS.PxE bits.
5. Set PxCMD.SUD bits, this will result in COMRESET to be issued to the all the enabled ports.

### 12.4.3 Interrupts

#### 12.4.3.1 Interaction Between INTx# and MSI/MSI-X

When MSI is enabled via MMC.MSIE=1, INTx# interrupt is disabled regardless of the setting of the PCI CMD.ID bit. The state of the CMD.ID bit does not affect MSI and MSI-X generation. This is shown the table below.

**Table 12-4. Interaction Between INTx# and MSI**

| CMD.ID | MMC.MSIE/MXC.MXE | INTX# Assertion | MSI/MSI-X Generation |
|---|---|---|---|
| 0 | 0 | Yes | No |
| 0 | 1 | No | Yes |
| 1 | 0 | No | No |
| 1 | 1 | No | Yes |

### 12.4.4 Power Management

The SATA Host Controller supports the PCI specification and the SATA PCI specification defined power states. See section 8 of the *Serial ATA Advanced Host Controller Interface (AHCI) Specification, Rev 1.3.1*. This controller also implements specific power management like clock and power gating.

Once BIOS enables the clock gating during the boot process, this is fully hardware autonomous and requires no software intervention.

### 12.4.5 Automatic Transition of Partial to Slumber

This feature is supported per the AHCI specification.

### 12.4.6 Enclosure Management and Activity LED

The Enclosure management is supported per the *Serial ATA Advanced Host Controller Interface (AHCI) Specification, Rev 1.3.1*. LED support is per the SFF 8485 specification.

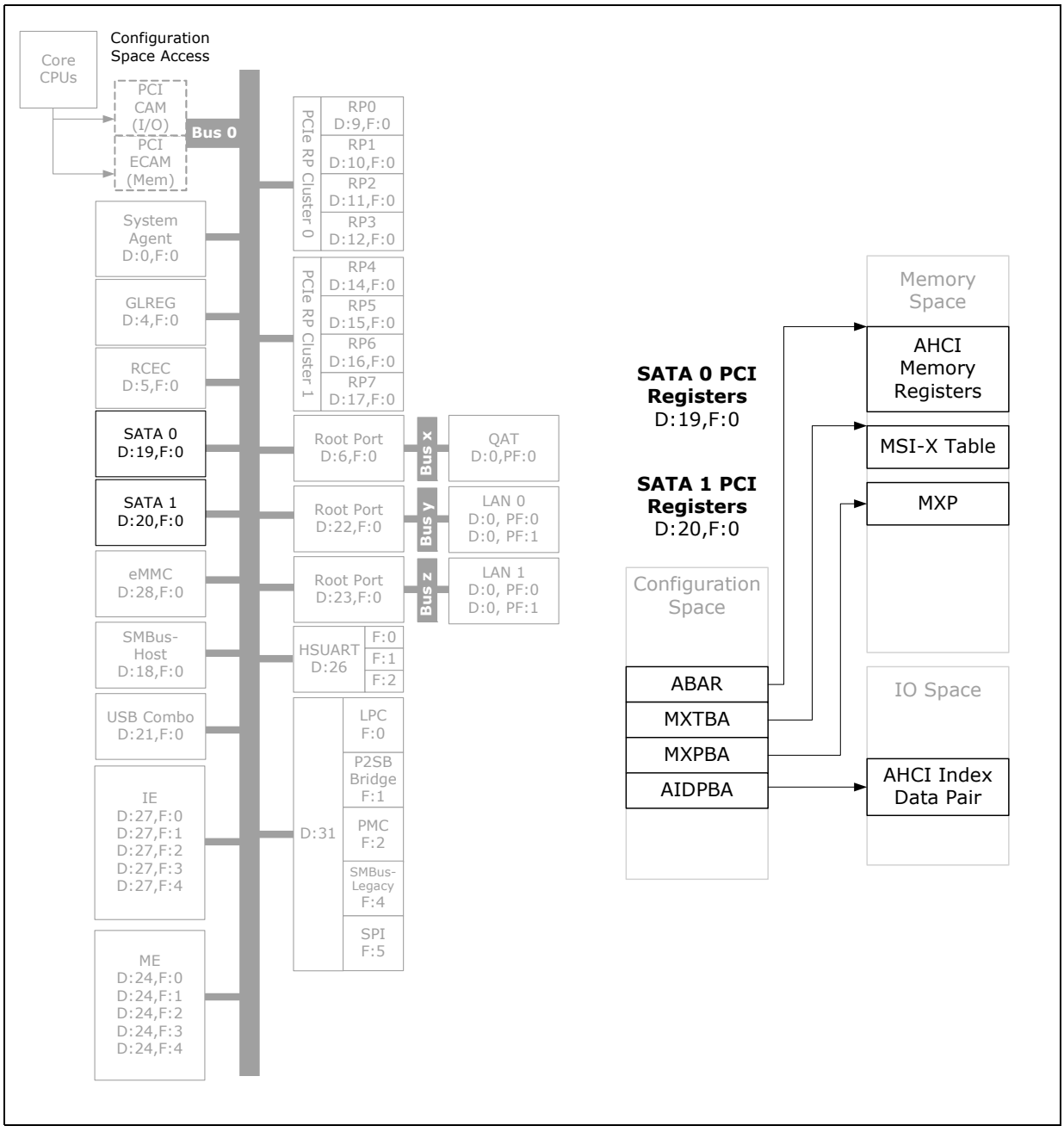
The SAF-TE, SES-2 and SGPIO message formats are defined in the corresponding specifications.

*Note:* When SGPIO is enabled it will always send 8 ports worth of messages.

## 12.5 Register Map

Figure 12-2 shows the associated registers from a system software viewpoint.

**Figure 12-2. Register Map**



### 12.5.1 PCI Configuration and Capabilities

**Table 12-5. Configuration and Capabilities Registers (Sheet 1 of 2)**

| CFG Address | Default | Instance Name | Name |
|---|---|---|---|
| 0x0 | 1C02_8086h | ID | Identifiers |
| 0x4 | 0000h | CMD | Command |
| 0x6 | 02B0h | STS | Device Status |
| 0x8 | 00h | RID | Revision ID |
| 0x9 | 01h | PI | Programming Interface |
| 0xA | 0106h | CC | Class Code |
| 0xC | 00h | CLS | Cache Line Size |
| 0xD | 00h | MLT | Master Latency Timer |
| 0xE | 00h | HTYPE | Header Type |
| 0x10 | 0000_0000h | MXTBA | MSI-X Table Base Address |
| 0x14 | 0000_0000h | MXPBA | MXP Base Address |
| 0x20 | 0000_0001h | AIDPBA | AHCI Index Data Pair Base Address |
| 0x24 | 0000_0000h | ABAR | AHCI Base Address |
| 0x2C | 0000_0000h | SS | Sub System Identifiers |
| 0x34 | 80h | CAP | Capabilities Pointer |
| 0x3C | 0100h | INTR | Interrupt Information |
| 0x70 | A801h | PID | PCI Power Management Capability ID |
| 0x72 | 4003h | PC | PCI Power Management Capabilities |
| 0x74 | 0008h | PMCS | PCI Power Management Control and Status |
| 0x80 | 7005h | MID | Message Signaled Interrupt Identifier |
| 0x82 | 0000h | MC | Message Signaled Interrupt Message Control |
| 0x84 | 0000_0000h | MA | Message Signaled Interrupt Message Address |
| 0x88 | 0000h | MD | Message Signaled Interrupt Message Data |
| 0x90 | 0000_0000h | MAP | Port Mapping Register |
| 0x94 | 0000_0000h | PCS | Port Control and Status |
| 0x9C | 0000_0000h | SATAGC | SATA General Configuration |
| 0xA0 | 00h | SIRI | SATA Initialization Register Index |
| 0xA4 | 0000_0000h | SIRD | SATA Initialization Register Data |
| 0xA8 | 0010_0012h | SATACR0 | Serial ATA Capability Register 0 |
| 0xAC | 0000_0048h | SATACR1 | Serial ATA Capability Register 1 |
| 0xB0 | 0013h | FLRCID | FLR Capability ID |

### Table 12-5. Configuration and Capabilities Registers (Sheet 2 of 2)

| | | | |
|---|---|---|---|
| 0xB2 | 0306h | FLRCAP | FLR Capability Length and Version |
| 0xB4 | 0000h | FLRCTL | FLR Control |
| 0xC0 | 0000_0000h | SP | Scratch Pad |
| 0xD0 | 0011h | MXID | MSI-X Identifiers |
| 0xD2 | 0000h | MXC | MSI-X Message Control |
| 0xD4 | 0000_0000h | MXT | MSI-X Table Offset / Table BIR |
| 0xD8 | 0000_0000h | MXP | MSI-X PBA Offset / PBA BIR |
| 0xE0 | 0000_0000h | BFCS | BIST FIS Control/Status |
| 0xE4 | 0000_0000h | BFTD1 | BIST FIS Transmit Data 1 |
| 0xE8 | 0000_0000h | BFTD2 | BIST FIS Transmit Data 2 |
| 0xF8 | 0800_0FB1h | MFID | Manufacturing ID |

## 12.5.2 Memory-Mapped Registers

**Table 12-6. Memory Mapped Registers (Sheet 1 of 5)**

| MEM Address | Default | Instance Name | Name |
|---|---|---|---|
| **MEM BaseAddress: ABAR** | | | |
| 0x0 | FF36_FF07h | GHC_CAP | HBA Capabilities |
| 0x4 | 8000_0000h | GHC | Global HBA Control |
| 0x8 | 0000_0000h | IS | Interrupt Status Register |
| 0xC | 0000_0000h | GHC_PI | Ports Implemented |
| 0x10 | 0001_0300h | VS | AHCI Version |
| 0x1C | 0160_0002h | EM_LOC | Enclosure Management Location |
| 0x20 | 0701_0000h | EM_CTL | Enclosure Management Control |
| 0x24 | 0000_003Ch | GHC_CAP2 | HBA Capabilities Extended |
| 0xA0 | 0000_0048h | VSP | Vendor Specific |
| 0xA4 | 0010_02DEh | VS_CAP | Vendor Specific Capabilities Register |
| 0xC0 | 0031_1C02h | RPID | RAID Platform ID |
| 0xC4 | 0000h | PFB | Premium Feature Block |
| 0xC8 | 003Fh | SFM | SW Feature Mask |
| 0x100 | 0000_0000h | PxCLB0 | Port [0-7] Command List Base Address |
| 0x104 | 0000_0000h | PxCLBU0 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x108 | 0000_0000h | PxFB0 | Port [0-7] FIS Base Address |
| 0x10C | 0000_0000h | PxFBU0 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x110 | 0000_0000h | PxIS0 | Port [0-7] Interrupt Status |
| 0x114 | 0000_0000h | PxIE0 | Port [0-7] Interrupt Enable |
| 0x118 | 0000_0004h | PxCMD0 | Port [0-7] Command |
| 0x120 | 0000_007Fh | PxTFD0 | Port [0-7] Task File Data |
| 0x124 | FFFF_FFFFh | PxSIG0 | Port [0-7] Signature |
| 0x128 | 0000_0000h | PxSSTS0 | Port [0-7] Serial ATA Status |
| 0x12C | 0000_0000h | PxSCTL0 | Port [0-7] Serial ATA Control |
| 0x130 | 0000_0000h | PxSERR0 | Port [0-7] Serial ATA Error |
| 0x134 | 0000_0000h | PxSACT0 | Port [0-7] Serial ATA Active |
| 0x138 | 0000_0000h | PxCI0 | Port [0-7] Commands Issued |
| 0x13C | 0000_0000h | PxSNTF0 | Port [0-7] SNotification |
| 0x144 | 1E02_2852h | PxDEVSLP0 | Port [0-7] Device Sleep |
| 0x180 | 0000_0000h | PxCLB1 | Port [0-7] Command List Base Address |

### Table 12-6. Memory Mapped Registers (Sheet 2 of 5)

| | | | |
|---|---|---|---|
| 0x184 | 0000_0000h | PxCLBU1 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x188 | 0000_0000h | PxFB1 | Port [0-7] FIS Base Address |
| 0x18C | 0000_0000h | PxFBU1 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x190 | 0000_0000h | PxIS1 | Port [0-7] Interrupt Status |
| 0x194 | 0000_0000h | PxIE1 | Port [0-7] Interrupt Enable |
| 0x198 | 0000_0004h | PxCMD1 | Port [0-7] Command |
| 0x1A0 | 0000_007Fh | PxTFD1 | Port [0-7] Task File Data |
| 0x1A4 | FFFF_FFFFh | PxSIG1 | Port [0-7] Signature |
| 0x1A8 | 0000_0000h | PxSSTS1 | Port [0-7] Serial ATA Status |
| 0x1AC | 0000_0000h | PxSCTL1 | Port [0-7] Serial ATA Control |
| 0x1B0 | 0000_0000h | PxSERR1 | Port [0-7] Serial ATA Error |
| 0x1B4 | 0000_0000h | PxSACT1 | Port [0-7] Serial ATA Active |
| 0x1B8 | 0000_0000h | PxCI1 | Port [0-7] Commands Issued |
| 0x1BC | 0000_0000h | PxSNTF1 | Port [0-7] SNotification |
| 0x1C4 | 1E02_2852h | PxDEVSLP1 | Port [0-7] Device Sleep |
| 0x200 | 0000_0000h | PxCLB2 | Port [0-7] Command List Base Address |
| 0x204 | 0000_0000h | PxCLBU2 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x208 | 0000_0000h | PxFB2 | Port [0-7] FIS Base Address |
| 0x20C | 0000_0000h | PxFBU2 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x210 | 0000_0000h | PxIS2 | Port [0-7] Interrupt Status |
| 0x214 | 0000_0000h | PxIE2 | Port [0-7] Interrupt Enable |
| 0x218 | 0000_0004h | PxCMD2 | Port [0-7] Command |
| 0x220 | 0000_007Fh | PxTFD2 | Port [0-7] Task File Data |
| 0x224 | FFFF_FFFFh | PxSIG2 | Port [0-7] Signature |
| 0x228 | 0000_0000h | PxSSTS2 | Port [0-7] Serial ATA Status |
| 0x22C | 0000_0000h | PxSCTL2 | Port [0-7] Serial ATA Control |
| 0x230 | 0000_0000h | PxSERR2 | Port [0-7] Serial ATA Error |
| 0x234 | 0000_0000h | PxSACT2 | Port [0-7] Serial ATA Active |
| 0x238 | 0000_0000h | PxCI2 | Port [0-7] Commands Issued |
| 0x23C | 0000_0000h | PxSNTF2 | Port [0-7] SNotification |
| 0x244 | 1E02_2852h | PxDEVSLP2 | Port [0-7] Device Sleep |
| 0x280 | 0000_0000h | PxCLB3 | Port [0-7] Command List Base Address |
| 0x284 | 0000_0000h | PxCLBU3 | Port [0-7] Command List Base Address Upper 32-bits |

**Table 12-6. Memory Mapped Registers (Sheet 3 of 5)**

| | | | |
|---|---|---|---|
| 0x288 | 0000_0000h | PxFB3 | Port [0-7] FIS Base Address |
| 0x28C | 0000_0000h | PxFBU3 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x290 | 0000_0000h | PxIS3 | Port [0-7] Interrupt Status |
| 0x294 | 0000_0000h | PxIE3 | Port [0-7] Interrupt Enable |
| 0x298 | 0000_0004h | PxCMD3 | Port [0-7] Command |
| 0x2A0 | 0000_007Fh | PxTFD3 | Port [0-7] Task File Data |
| 0x2A4 | FFFF_FFFFh | PxSIG3 | Port [0-7] Signature |
| 0x2A8 | 0000_0000h | PxSSTS3 | Port [0-7] Serial ATA Status |
| 0x2AC | 0000_0000h | PxSCTL3 | Port [0-7] Serial ATA Control |
| 0x2B0 | 0000_0000h | PxSERR3 | Port [0-7] Serial ATA Error |
| 0x2B4 | 0000_0000h | PxSACT3 | Port [0-7] Serial ATA Active |
| 0x2B8 | 0000_0000h | PxCI3 | Port [0-7] Commands Issued |
| 0x2BC | 0000_0000h | PxSNTF3 | Port [0-7] SNotification |
| 0x2C4 | 1E02_2852h | PxDEVSLP3 | Port [0-7] Device Sleep |
| 0x300 | 0000_0000h | PxCLB4 | Port [0-7] Command List Base Address |
| 0x304 | 0000_0000h | PxCLBU4 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x308 | 0000_0000h | PxFB4 | Port [0-7] FIS Base Address |
| 0x30C | 0000_0000h | PxFBU4 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x310 | 0000_0000h | PxIS4 | Port [0-7] Interrupt Status |
| 0x314 | 0000_0000h | PxIE4 | Port [0-7] Interrupt Enable |
| 0x318 | 0000_0004h | PxCMD4 | Port [0-7] Command |
| 0x320 | 0000_007Fh | PxTFD4 | Port [0-7] Task File Data |
| 0x324 | FFFF_FFFFh | PxSIG4 | Port [0-7] Signature |
| 0x328 | 0000_0000h | PxSSTS4 | Port [0-7] Serial ATA Status |
| 0x32C | 0000_0000h | PxSCTL4 | Port [0-7] Serial ATA Control |
| 0x330 | 0000_0000h | PxSERR4 | Port [0-7] Serial ATA Error |
| 0x334 | 0000_0000h | PxSACT4 | Port [0-7] Serial ATA Active |
| 0x338 | 0000_0000h | PxCI4 | Port [0-7] Commands Issued |
| 0x33C | 0000_0000h | PxSNTF4 | Port [0-7] SNotification |
| 0x344 | 1E02_2852h | PxDEVSLP4 | Port [0-7] Device Sleep |
| 0x380 | 0000_0000h | PxCLB5 | Port [0-7] Command List Base Address |
| 0x384 | 0000_0000h | PxCLBU5 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x388 | 0000_0000h | PxFB5 | Port [0-7] FIS Base Address |

### Table 12-6. Memory Mapped Registers (Sheet 4 of 5)

| | | | |
|---|---|---|---|
| 0x38C | 0000_0000h | PxFBU5 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x390 | 0000_0000h | PxIS5 | Port [0-7] Interrupt Status |
| 0x394 | 0000_0000h | PxIE5 | Port [0-7] Interrupt Enable |
| 0x398 | 0000_0004h | PxCMD5 | Port [0-7] Command |
| 0x3A0 | 0000_007Fh | PxTFD5 | Port [0-7] Task File Data |
| 0x3A4 | FFFF_FFFFh | PxSIG5 | Port [0-7] Signature |
| 0x3A8 | 0000_0000h | PxSSTS5 | Port [0-7] Serial ATA Status |
| 0x3AC | 0000_0000h | PxSCTL5 | Port [0-7] Serial ATA Control |
| 0x3B0 | 0000_0000h | PxSERR5 | Port [0-7] Serial ATA Error |
| 0x3B4 | 0000_0000h | PxSACT5 | Port [0-7] Serial ATA Active |
| 0x3B8 | 0000_0000h | PxCI5 | Port [0-7] Commands Issued |
| 0x3BC | 0000_0000h | PxSNTF5 | Port [0-7] SNotification |
| 0x3C4 | 1E02_2852h | PxDEVSLP5 | Port [0-7] Device Sleep |
| 0x400 | 0000_0000h | PxCLB6 | Port [0-7] Command List Base Address |
| 0x404 | 0000_0000h | PxCLBU6 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x408 | 0000_0000h | PxFB6 | Port [0-7] FIS Base Address |
| 0x40C | 0000_0000h | PxFBU6 | Port [0-7] FIS Base Address Upper 32-bits |
| 0x410 | 0000_0000h | PxIS6 | Port [0-7] Interrupt Status |
| 0x414 | 0000_0000h | PxIE6 | Port [0-7] Interrupt Enable |
| 0x418 | 0000_0004h | PxCMD6 | Port [0-7] Command |
| 0x420 | 0000_007Fh | PxTFD6 | Port [0-7] Task File Data |
| 0x424 | FFFF_FFFFh | PxSIG6 | Port [0-7] Signature |
| 0x428 | 0000_0000h | PxSSTS6 | Port [0-7] Serial ATA Status |
| 0x42C | 0000_0000h | PxSCTL6 | Port [0-7] Serial ATA Control |
| 0x430 | 0000_0000h | PxSERR6 | Port [0-7] Serial ATA Error |
| 0x434 | 0000_0000h | PxSACT6 | Port [0-7] Serial ATA Active |
| 0x438 | 0000_0000h | PxCI6 | Port [0-7] Commands Issued |
| 0x43C | 0000_0000h | PxSNTF6 | Port [0-7] SNotification |
| 0x444 | 1E02_2852h | PxDEVSLP6 | Port [0-7] Device Sleep |
| 0x480 | 0000_0000h | PxCLB7 | Port [0-7] Command List Base Address |
| 0x484 | 0000_0000h | PxCLBU7 | Port [0-7] Command List Base Address Upper 32-bits |
| 0x488 | 0000_0000h | PxFB7 | Port [0-7] FIS Base Address |
| 0x48C | 0000_0000h | PxFBU7 | Port [0-7] FIS Base Address Upper 32-bits |

**Table 12-6. Memory Mapped Registers (Sheet 5 of 5)**

| | | | |
|---|---|---|---|
| 0x490 | 0000_0000h | PxIS7 | Port [0-7] Interrupt Status |
| 0x494 | 0000_0000h | PxIE7 | Port [0-7] Interrupt Enable |
| 0x498 | 0000_0004h | PxCMD7 | Port [0-7] Command |
| 0x4A0 | 0000_007Fh | PxTFD7 | Port [0-7] Task File Data |
| 0x4A4 | FFFF_FFFFh | PxSIG7 | Port [0-7] Signature |
| 0x4A8 | 0000_0000h | PxSSTS7 | Port [0-7] Serial ATA Status |
| 0x4AC | 0000_0000h | PxSCTL7 | Port [0-7] Serial ATA Control |
| 0x4B0 | 0000_0000h | PxSERR7 | Port [0-7] Serial ATA Error |
| 0x4B4 | 0000_0000h | PxSACT7 | Port [0-7] Serial ATA Active |
| 0x4B8 | 0000_0000h | PxCI7 | Port [0-7] Commands Issued |
| 0x4BC | 0000_0000h | PxSNTF7 | Port [0-7] SNotification |
| 0x4C4 | 1E02_2852h | PxDEVSLP7 | Port [0-7] Device Sleep |
| 0x580 | 0000_0000h | EM_MF | Enclosure Management Message Format |
| 0x584 | 0000_0000h | EM_LED | Enclosure Management LED |
| **MEM BaseAddress: MXTBA** | | | |
| 0x0 | 0000_0000h | MXTEnMLA | MSI-X Table Entries 0 Message Lower Address |
| 0x4 | 0000_0000h | MXTEnMUA | MSI-X Table Entries 0 Message Upper Address |
| 0x8 | 0000_0000h | MXTEnMD | MSI-X Table Entries 0 Message Data |
| 0xC | 0000_0001h | MXTEnVC | MSI-X Table Entries 0 Vector Control |
| **MEM BaseAddress: MXPBA** | | | |
| 0x0 | 0000_0000h | MXPQW0_DW0 | MSI-X Pending Bit Array QW 0 |
| 0x4 | 0000_0000h | MXPQW0_DW1 | MSI-X Pending Bit Array QW 1 |

## 12.5.3 I/O-Mapped Registers

### Table 12-7. I/O Mapped Registers

| IO Address | Default | Instance Name | Name |
|---|---|---|---|
| **IO BaseAddress: AIDPBA** | | | |
| 0x10 | 0000_0000h | INDEX | AHCI Index Register |
| 0x14 | 0000_0000h | DATA | AHCI Data Register |

§ §

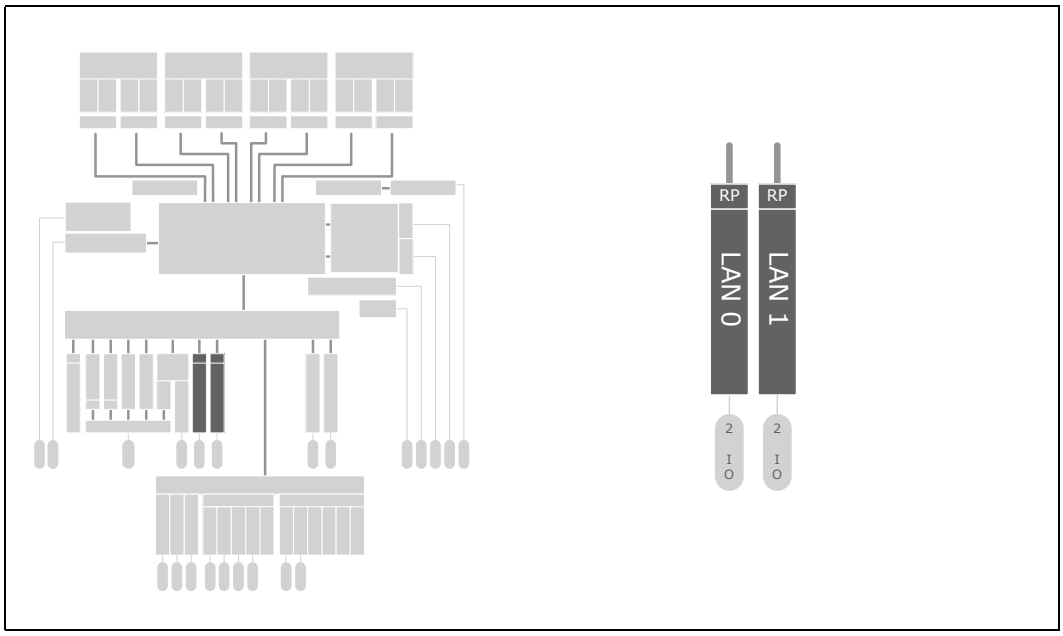
# 13 LAN Controllers

There are two SoC LAN controllers (LAN0 and LAN1) with two ports each, providing a total of four ports. Each port contains an independent LAN Media Access Control (MAC) that supports a XGMII-like interface link to an internal PHY (see Figure 13-1).

The internal MAC/PHY combine to support 1GbE (KX and SGMII), 2.5GbE and 10GbE operation through various modes of operation.

Refer to Section 13.2, "Supported Modes of Operation" on page 415 for all supported modes of Ethernet for use with internal PHYs (native support) and external PHYs.

**Figure 13-1. What is Covered in This Chapter**



## Table 13-1. References

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| DMTF | 1.0 | May 2009 | *DMTF Network Controller Sideband Interface (NC-SI) Specification)* |
| IEEE | | | *IEEE 1149.6 standard for Boundary Scan* (MDI pins excluded) |
| IEEE | | | *IEEE 802.3-2012*, Clauses 70 through 74. |
| IEEE | | 2001 | *IEEE standard 1149.1*, 2001 Edition (JTAG). Institute of Electrical and Electronics Engineers (IEEE). |
| IEEE | | | *IEEE standard 802*.1Q for VLAN |
| IEEE | | September 2004 | *IEEE 1588*, International Standard, Precision clock synchronization protocol for networked measurement and control systems (1588V1/1588V2) |
| IEEE | | January 19, 2006 | *IEEE P802.1AE/D5.1*, Media Access Control (MAC) Security |
| IEEE | | | *IEEE 802.3-2012*, Clause 78 |
| IETF | | | *IETF IPv4 Specification*, (RFC 791) |
| IETF | | | *IETF IPv6 Specification*, (RFC 2460) |
| IETF | | | *IETF TCP Specification*, (RFC 793) |
| IETF | | | *IETF UDP Specification*, (RFC 768) |
| IETF | | | *IETF ARP Specification*, (RFC 826) |
| PCI-SIG | | | *PCI Express Base Specification*, Revision 3.0 |
| PCI-SIG | 1.2 | March 2004 | *PCI Bus Power Management Interface Specification*, Revision 1.2 |
| PCI-SIG | 2.0 | | PICMG 3.1 |
| PCI-SIG | 1.1 | September 8, 2009 | *PCI Single Root I/O Virtualization and Sharing Specification* |
| SMBus | 2.0 | August 2000 | *System Management Bus (SMBus) Specification*, SBS Implementers Forum |

## 13.1 Signal Descriptions

The signal descriptions are shown in Table 13-2, "Signal Names and Descriptions". For additional signal information, including whether a signal is assigned to a shared-function pin or can be used as a Customer GPIO signal see Section 39.1, "Directory of Signal Names and Pin Names". The Direction/Type column of Table 13-2, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.
- Differential = Differential signal pair. Two pins: Positive; Negative.

**Table 13-2. Signal Names and Descriptions (Sheet 1 of 3)**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| **LAN Ports 0 and 1 - LAN Controller 0** | | | |
| LAN0_PORT0_I2C_CLK | I,O-OD | Yes | **$I^2C$ Clock:** The 2-wire management interface Clock used to access the management registers of an external optical module of port n. One clock pulse is generated for each data bit transferred. |
| LAN0_PORT0_I2C_DATA | I,O-OD | Yes | **$I^2C$ Data:** The Data of the 2-wire management interface used to access the management registers of an external optical module of port n. Stable during the clock high period (unless it is a start or stop condition). |
| LAN0_PORT0_LED[3:0] | O-OD | Yes | **LED Drivers:** Four LED drivers for this LAN Port. |
| LAN0_PORT0_SDP[3:0] | I,O | Yes | **Software Defined Pins:** Four general-purpose SDPs for this LAN Port, can be used to support IEEE1588 auxiliary devices or input for external interrupts. |
| LAN0_PORT0_RX_DP<br>LAN0_PORT0_RX_DN | I, Differential | No | **PHY Differential Input:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN0_PORT0_TX_DP<br>LAN0_PORT0_TX_DN | O, Differential | No | **PHY Differential Output:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN0_PORT1_I2C_CLK | I,O-OD | Yes | **$I^2C$ Clock:** The 2-wire management interface Clock used to access the management registers of an external optical module of port n. One clock pulse is generated for each data bit transferred. |
| LAN0_PORT1_I2C_DATA | I,O-OD | Yes | **$I^2C$ Data:** The Data of the 2-wire management interface used to access the management registers of an external optical module of port n. Stable during the high period of the clock (unless it is a start or stop condition). |
| LAN0_PORT1_LED[1:0] | O-OD | Yes | **LED Drivers:** Two LED drivers for this LAN Port. |
| LAN0_PORT1_SDP[3:0] | I,O | Yes | **Software Defined Pins:** Four general-purpose SDPs for this LAN Port. Can be used to support IEEE1588 auxiliary devices or input for external interrupts. |
| LAN0_PORT1_RX_DP<br>LAN0_PORT1_RX_DN | I | No | **PHY Differential Input:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G,100 Mb/s full duplex, 10 Mb/s full duplex, and SFI. |
| LAN0_PORT1_TX_DP<br>LAN0_PORT1_TX_DN | O | No | **PHY Differential Output:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G,100 Mb/s full duplex, 10 Mb/s full duplex, and SFI. |
| LAN0_RBIAS | I,O | No | **Bias Resistor:** External resistor bias input. The platform board must provide a 1000 Ω, 0.10% resistor connected from this pin to VCCKR0LCPLL_LV. Intel recommends shielding the trace to this pin. |

## Table 13-2. Signal Names and Descriptions (Sheet 2 of 3)

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| **LAN Ports 0 and 1 - LAN Controller 1** | | | |
| LAN1_PORT0_I2C_CLK | I,O-OD | Yes | **I²C Clock:** The Clock of the 2-wire management interface used to access the management registers of an external optical module of port n.<br>One clock pulse is generated for each data bit transferred. |
| LAN1_PORT0_I2C_DATA | I,O-OD | Yes | **I²C Data:** The Data of the 2-wire management interface used to access the management registers of an external optical module of port n.<br>Stable during the high period of the clock (unless it is a start or stop condition). |
| LAN1_PORT0_LED[1:0] | O-OD | Yes | **LED Drivers:** Two LED drivers for this LAN Port. |
| LAN1_PORT0_SDP[1:0] | I,O | Yes | **Software Defined Pins:** Two general-purpose SDPs for this LAN Port. Can be used to support IEEE1588 auxiliary devices or input for external interrupts. |
| LAN1_PORT0_RX_DP<br>LAN1_PORT0_RX_DN | I, Differential | No | **PHY Differential Input:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN1_PORT0_TX_DP<br>LAN1_PORT0_TX_DN | O, Differential | No | **PHY Differential Output:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN1_PORT1_I2C_CLK | I,O-OD | Yes | **I²C Clock:** The Clock of the 2-wire management interface used to access the management registers of an external optical module of port n.<br>One clock pulse is generated for each data bit transferred. |
| LAN1_PORT1_I2C_DATA | I,O-OD | Yes | **I²C Data:** 2-wire management interface data used to access the management registers of an external optical module of port n.<br>Stable during high period of clock (unless it is a start or stop condition). |
| LAN1_PORT1_LED[1:0] | O-OD | Yes | **LED Drivers:** Two LED drivers for this LAN Port. |
| LAN1_PORT1_SDP[1:0] | I,O | Yes | **Software Defined Pins:** Two general-purpose SDPs for this LAN Port, can be used to support IEEE1588 auxiliary devices or input for external interrupts. |
| LAN1_PORT1_RX_DP<br>LAN1_PORT1_RX_DN | I, Differential | No | **PHY Differential Input:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN1_PORT1_TX_DP<br>LAN1_PORT1_TX_DN | O, Differential | No | **PHY Differential Output:** This lane is used in 10GBASE-KR, 1000BASE-KX, 2.5G and SFI. |
| LAN1_RBIAS | I,O | No | **Bias Resistor:** External resistor bias input. The platform board must provide a 1000 Ω, 0.10% resistor connected from this pin to VCCKR1LCPLL_LV. Intel recommends shielding the trace to this pin. |
| **Signals Common to LAN Controllers 0 and 1** | | | |
| LAN_MDC | O | Yes | **Management Clock:** Clock output for accessing the external PHYs management registers. |
| LAN_MDIO | I,O-OD | Yes | **Management Data:** Bi-directional signal for serial data transfers between MAC and the external PHY. |
| SMB_LAN_DATA | I,O-OD | Yes | **SMBus Data:** Stable during the high period of the clock (unless it is a start or stop condition). |
| SMB_LAN_CLK | I,O-OD | Yes | **SMBus Clock:** One clock pulse is generated per data bit transferred. |
| SMB_LAN_ALRT_N | I,O-OD | Yes | **SMBus Alert:** Acts as an interrupt pin of a slave device on the SMBus. |
| NCSI_ARB_IN | I | Yes | **NC-SI Arbitration In:** Along with NCSI_ARB_OUT, this signal provides hardware-based arbitration when multiple Network Controllers are used. The SoC provides an internal 20-kΩ pull-down resistor. |
| NCSI_ARB_OUT | O | Yes | **NC-SI Arbitration Out:** Along with NCSI_ARB_IN, this signal provides hardware-based arbitration when multiple Network Controllers are used. |

## Table 13-2. Signal Names and Descriptions (Sheet 3 of 3)

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| **Signals Common to LAN Controllers 0 and 1 (continued)** | | | |
| NCSI_CLK_IN | I | Yes | **NC-SI Reference Clock Input:** Synchronous clock reference for receive, transmit, and control interface. It is a 50 MHz clock ±100 ppm. The SoC provides an internal 20-kΩ pull-up resistor. |
| NCSI_CRS_DV | O-Tri | Yes | **Carrier Sense/Receive Data Valid:** Tri-state output signal to the MC. Indicates that the data transmitted from the SoC to the MC is valid. If this signal is used, the platform board must provide a pull-up resistor when the output signal is in the high-impedance state. |
| NCSI_RXD[1:0] | O-Tri | Yes | **MC Receive Data:** Tri-state output data signals from the SoC LAN controllers to the MC. The SoC provides an internal 20-kΩ pull-up resistor for each output signal. |
| NCSI_TX_EN | I | Yes | **MC Transmit Enable:** Indicates received data from the MC is valid. |
| NCSI_TXD[1:0] | I | Yes | **MC Transmit Data:** Data signals from the MC to the SoC LAN controllers. |

***Note:*** The SoC LAN controllers only have one SMBus and one NC-SI bus associated with all LAN ports. Both the SMBus and NC-SI can be used simultaneously by the platform board design provided that both sets of signals are configured.

## 13.2 Supported Modes of Operation

The SoC LAN controllers support speeds of 2.5 GbE[1], 1 GbE, and 10 GbE, as well as 10/100 Mb/s in only full-duplex mode.

There are two SoC LAN controllers (LAN0 and LAN1) with two ports each, providing a total of four ports.

LAN Controller 0 will be enabled for product SKU that only has two ports.

For product SKU with LAN controller 0 and LAN controller 1 if SOC Ethernet is used LAN controller 0 must be enabled.

For each port, there are options to use internal PHYs supporting:

- 10GBASE-KR
- SFI
- 1000BASE-KX
- 2500BASE-X

There are also options to connect to external PHYs via:

- 10GBASE-KR
- 1000BASE-KX
- SGMII is supported for Full Duplex 10 Mb/s, 100 Mb/s and 1 GbE

Each integrated I/O interface block provides the following physical interfaces and electrical modes:

- 10GbE PHY supports:
  - 10GBASE-KR for backplane applications (IEEE802.3 clause 72)
  - 10GBASE-KR FEC (IEEE 802.3 Clause 74)
  - 1000BASE-KX for backplane applications (IEEE802.3 clause 70)
  - Auto-negotiation for backplane Ethernet (IEEE 802.3 Clause 73)
  - SFI compatible interface to: SFP+[2] (SFF8431)
  - KR to 10GBASE-T PHY (Intel® X557-AT2/AT4 LAN PHY)

*Note:* No support for BASE-T SFP modules.

*Note:* No support for half duplex at 10 Mb/s, 100 Mb/s, 1 GbE, 2.5 GbE or 10 GbE.

*Note:* LAN port topologies must be the same for each respective LAN, for example, Ports 0 and 1 of LAN0 must be the same Ethernet mode. However, LAN 0 and LAN 1 can support different topologies as long as care is given to the trace layout. Due to the cross-talk requirements, there is no mixing of DAC (Direct-Attached-Cable) and Optical modules on ports 0 and 1 of the same LAN when using an SFI (native) design.

1. Auto-negotiation is not supported in 2.5 GbE. Auto-negotiate is supported in all other modes.
2. Some system designs might require adding a InPhi* CS4227(dual port)/CS4223 (quad port) re-timer PHY between the SoC and the SFP+ module see Table 13-3, "Supported System Configurations".

Table 13-3, "Supported System Configurations" lists all the supported system configurations, operating modes, and link partners for the LAN controllers. Selecting the desired PHY and system configuration is determined at power on through the use of the shared SPI Flash configuration.

## Table 13-3. Supported System Configurations

| Connection | Speed | Electrical Interface | Third Party PHYs/Switches[4] |
|---|---|---|---|
| Backplane | 1 GbE | 1000BASE-KX | Intel® FM4000/FM5000/FM6000 Ethernet switches<br>Broadcom* BCM5684x-series switches |
| Backplane | 10 GbE | 10GBASE-KR | Intel FM5000/FM6000 Ethernet switches<br>Broadcom* BCM5684x-series switches |
| Backplane | 2.5 GbE | 2500BASE-X [1] | Intel FM4000/FM5000/FM6000 Ethernet switches<br>Broadcom* BCM5684x-series switches |
| 10BASE-T<br>100BASE-TX<br>1000BASE-T | 10 Mb/s<br>100 Mb/s<br>1 GbE | SGMII (no half duplex support) | Marvell* 88E1514/88E1512 [5]<br>Marvell 88E1543 |
| 10GBASE-T [7] | 10 GbE | KR | Intel® X557-AT2/AT4 10 GbE PHY |
| 1000BASE-T [7] | 1 GbE | KX | Intel® X557-AT2/AT4 10 GbE PHY |
| SFP+ [7] | 10 GbE | SFI | Native SFI support |
| SFP+ [7] | 10 GbE | KR [3] | Inphi CS4227 (2 port) [6]<br>Inphi CS4223 (4 port) [6] |

***Notes:***
1. 2500BASE-X is not an IEEE standard. The SoC supports 2.5G rate only.
2. No support for half duplex.
3. Inphi devices required for long reach support in KR mode.
4. XFP+ devices are not supported and should not be used as they may impact the reliability of the LAN port.
5. The 88E1512 and 88E1514 only supports two MDIO addresses (0 or 1) therefore a maximum of 2 of these PHY devices can be used with each SoC.
6. CS422X devices include integrated Rx CAPS. DO NOT include 100nF for Rx traces close to the devices.
7. Supported only in 10G capable LAN controller SKU.

## 13.3 Feature List

### Table 13-4. Network Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Compliant with the 1 Gb/s Ethernet/802.3 (KX) Specification | Y | Y |
| Compliant with the 10 Gb/s 802.3 (KR) specification | Y | Y |
| Support of 10GBASE-KR FEC | Y | Y |
| Full-Duplex Operation at all Supported Speeds | Y | N |
| 802.3az Energy Efficient Ethernet (EEE) Support | N | N |
| Support Jumbo Frames of up to 15.5 KB | Y [1] | Y [1] |
| MDIO Interface Clause 45 | Y | N |
| Flow Control Support: Send/Receive Pause Frames and Receive FIFO Thresholds | Y | Y |
| Statistics for Management and RMON | Y | Y |
| 802.1q VLAN Support | Y | Y |
| Double VLAN | Y | Y |

1. The SoC LAN controllers support full-size 15.5 KB jumbo packets while in a basic mode of operation. When virtualization is enabled, or OS2BMC is enabled, then the SoC LAN controllers supports 9.5 KB jumbo packets. Packets to/from the MC longer than 2 KB are filtered out.

### Table 13-5. Host Interface Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Integrated I/O | Y | N |
| 64-bit Address Support for Systems Using More Than 4 GB of Physical Memory | Y | Y |
| Outstanding Requests for Tx Data Buffers | 16 | 16 |
| Outstanding Requests for Tx Descriptors | 8 | 8 |
| Outstanding Requests for Rx Descriptors | 8 | 8 |
| Credits for P-H/P-D/NP-H/NP-D (shared for the two ports) | 4/255/4/255 [1] | 16/16/4/4 |
| Credits for C-H/C-D (shared for the two ports) | 4/255 | N |
| Max Payload Size Supported | 64 - 512 bytes | 512 bytes |
| Max Request Size Supported | 64B | 2 KB |
| Vital Product Data (VPD) | Y | Y |
| End to End CRC (ECRC) | N/A | Y |
| Latency Tolerance Reporting (LTR) | N | N |
| ID-Based Ordering (IDO) | Y | N |
| Access Control Services (ACS) | Y | N |
| PCIe Functions Off through BIOS Control While LAN Ports Are On | Y | N |

1. Each data credit accounts for 16 bytes.

## Table 13-6. Miscellaneous Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Serial Flash Interface | N [1] | Y |
| Configurable LED Operation for Software or OEM Customization of LED Displays | Y | Y |
| Shared SPI Flash Space for Private Configuration | Y | Y |
| Device Disable Capability | Y | Y |
| Watchdog Timer | Y | Y |
| Time Sync (IEEE 1588 v2) | Y | Y |

1. Serial Flash interface is available at the SoC level and is shared among all the integrated devices.

## Table 13-7. LAN Functions Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Programmable Host Memory Receive Buffers | Y | Y |
| Descriptor Ring Management Hardware for Transmit and Receive | Y | Y |
| ACPI Register Set and Power Down Functionality Supporting D0 and D3 States | Y | Y |
| Software-Controlled Global Reset Bit (Resets Everything Except the Configuration Registers) | Y | Y |
| Software-Definable Pins (SDP) (per port) | 2 [1] | 8 |
| SDP Pins Can Be Configured as General Purpose Interrupts | Y | Y |
| Wake on LAN (WoL) | Y | Y |
| IPv6 Wake-up Filters | Y | Y |
| Configurable (through shared SPI Flash update) Wake-up Flexible Filters | Y | Y |
| Default Configuration by shared SPI Flash update for all LEDs for Pre-Driver Functionality | Y | Y |
| LAN Function Disable Capability | Y | Y |
| Programmable Memory Transmit Buffers | 160 KB / port | 160 KB / port |
| Programmable Memory Receive Buffers | 384 KB / port | 512 KB / port |

1. Available number is dependent on I/O MUXing and selected functionality.

## Table 13-8. LAN Performance Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| TCP/UDP Segmentation Offload | 256 KB in all modes | 256 KB in all modes |
| TSO Interleaving for Reduced Latency | Y | Y |
| TCP Receive Side Coalescing (RSC) | 32 flows / port | 32 flows / port |
| Rate Limit VM Tx Traffic per TC (i.e. per TxQ) | Y | N |
| IPv6 Support for IP/TCP and IP/UDP Receive Checksum Offload | Y | Y |
| Fragmented UDP Checksum Offload for Packet Reassembly | Y | Y |
| Message Signaled Interrupts (MSI) | Y | Y |
| Message Signaled Interrupts (MSI-X) | Y | Y |
| Interrupt Throttling Control to Limit Maximum Interrupt Rate and Improve CPU Use | Y | Y |
| Rx Packet Split Header | Y | Y |
| Multiple Rx Queues (RSS) | Y (multiple modes) | Y (multiple modes) |
| Flexible RSS | N | N |
| Flow Director Filters: up to 32 KB Flows by Hash Filters or up to 8 KB Perfect Match Filters | Y | Y |
| Number of Rx Queues (per port) | 128 | 128 |
| Number of Tx Queues (per port) | 128 | 128 |
| TCP Timer Interrupts<br>Relax Ordering | Y<br>Y | Y<br>Y |
| DMA Coalescing | Y | N |

### Table 13-9. Virtualization Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Support for Virtual Machine Device Queues (VMDq1 and Next Generation VMDq) | 64 | 64 |
| L2 Ethernet MAC Address Filters (Unicast and Multicast) | 128 | 128 |
| L2 VLAN Filters | 64 | 64 |
| PCI-SIG SR IOV | Y | Y |
| Multicast and Broadcast Packet Replication | Y | Y |
| Packet Mirroring | Y | Y |
| Packet Loopback | Y | Y |
| Traffic Shaping | Y | Y |
| Anti Spoof | MAC, VLAN, Ethertype | N |
| Forwarding Modes | MAC, VLAN E-tag | N |
| VEB Support | Y | N |

### Table 13-10. Manageability Features

| Feature | SoC LAN Controllers | 82599 |
|---|---|---|
| Advanced Pass Through-compatible Management Packet Transmit/Receive Support | Y | Y |
| SMBus Interface to an External MC | Y | Y |
| New Management Protocol Standards Support (NC-SI) Interface to an External MC | Y | Y |
| L2 Address Filters | 4 | 4 |
| VLAN L2 Filters | 8 | 8 |
| Flex L3 Port Filters | 16 | 16 |
| Flexible TCO Filters | 1 | 4 |
| L3 Address Filters (IPv4) | 4 | 4 |
| L3 Address Filters (IPv6) | 4 | 4 |
| Host-Based Application-to-BMC Network Communication Patch (OS2BMC) | Y | N |
| Flexible MAC Address | Y | N |
| MC Inventory of LOM Device Information | Y | N |
| iSCSI Boot Configuration Parameters via MC | Y | N |
| MC Monitoring | Y | N |
| NC-SI to MC | Y | N |
| NC-SI Arbitration | Y | N |
| NC-SI Package ID Via SDP Pins | Y | N |
| NC-SI Flow Control | Y | N |
| MCTP over SMBus (pass through and control) | Y | N |

## 13.4 Programmers Reference Manual

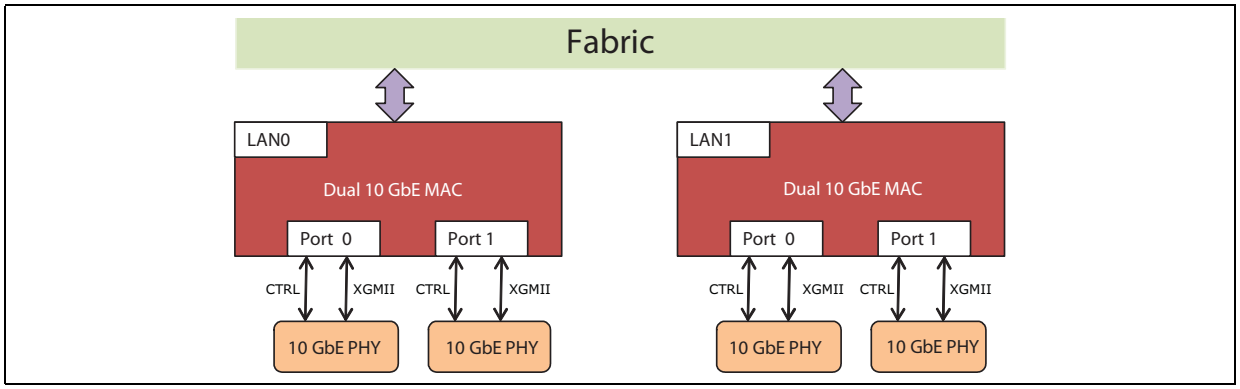
Much of the programming-related information and register descriptions for the SoC LAN controllers are in a separate document.

The descriptions of the MMIO registers and MSI-X registers located in memory space, the registers in I/O space, and the contents of the Shared NVM are described in the PRM. The SoC LAN controller registers located in PCI configuration space are provided in both the PRM and this document.

## 13.5 Architectural Overview

The SoC LAN controllers are a derivative of Intel® Ethernet Controller 82599 and largely supports most feature sets of this document.

**Figure 13-2. System Architecture and Interface**




*Note:* 10GbE PHY shown in Figure 13-2, "System Architecture and Interface" are internal to the SoC package.

### 13.5.1 PCIe Integrated Endpoint

From a systems view, the SoC LAN controllers are a PCIe* integrated endpoint consisting of two devices with two functions. Two independent controllers (LAN0 and LAN1), are connected to PCI Root Port with separate device and function numbers. LAN0 is connected to RP (D:22 F:0) 22 decimal while LAN1 is connected to RP (D:23 F:0) 23 decimal. Both ports on each controller are addressable by function 0 and 1.

SoC LAN controllers are highly configurable. Device IDs for each of the four functions are assigned, and its capabilities are enabled/disabled by data contained in the shared SPI Flash device. Configurable PCI and PCIe capabilities, and their offsets from the start of each capability location in PCI configuration space are listed in Table 13-11, "PCI and PCIe Capabilities Supported".

The LAN controllers are not connected through a standard PCIe bus. Though the LAN controllers appear as PCIe endpoints, and reports Gen1 speed and x1 width, the internal connection is not standard, and has much higher performance.

**Table 13-11. PCI and PCIe Capabilities Supported**

| Offset in Configuration Space (hexadecimal) | Type of Capability | Capability |
|---|---|---|
| 40 | PCI | PCI Power Management Interface (PMI) |
| 50 | PCI | Message Signaled Interrupts (MSI) |
| 70 | PCI | Message Signaled Interrupts, extended (MSI-X) |
| A0 | PCI | PCI Express Capability |
| E0 | PCI | Vital Product Data (VPD) |
| 100 | PCI Express Extended | Advanced Error Reporting (AER) |
| 140 | PCI Express Extended | Device Serial Number |
| 1B0 | PCI Express Extended | Access Control Services (ACS) |

*Note:* The dummy function device ID that is available on some Intel LAN components is not supported by the SoC.

Each of the four functions has six Base Address Registers (BARs) that are assigned address values during PCI bus enumeration. See Table 13-12, "Base Address Registers".

**Table 13-12. Base Address Registers**

| Offset in Configuration Space (hexadecimal) | 32-bit Register Name | Capability | Data Bytes Accessible in Given Space |
|---|---|---|---|
| 10 | BASE_ADDR_0 | 64-bit BAR for Memory Space for Memory-Mapped I/O (MMIOs) | 128 K |
| 14 | BASE_ADDR_1 | | |
| 18 | BASE_ADDR_2 | 32-bit BAR for I/O Space | 32 |
| 1C | BASE_ADDR_3 | Reserved | |
| 20 | BASE_ADDR_4 | 64-bit BAR for Memory Space for Message Signaled Interrupts, extended (MSI-X) | 16 K |
| 24 | BASE_ADDR_5 | | |

There are two device-specific registers located in configuration space for each function. They provide one of the two possible methods for accessing the SoC LAN controller internal command and status registers. This first method uses PCI configuration space of each of the four functions:

- 0x98 - IOADDR (32 bits)
- 0x9C - IODATA (32 bits)

The other method uses the system I/O space. Once the BASE_ADDR_2 BAR is enumerated for a particular function, it can be used to access the IOADDR and IODATA registers:

- BASE_ADDR_2 (I/O base address) plus 0 - IOADDR (32 bits)
- BASE_ADDR_2 (I/O base address) plus 4 - IODATA (32 bits)

*Note:* There are many configuration options that affect the parameter assignments and operation of the SoC LAN controllers. This document only provides an overview and does not show all possibilities available to the SoC customer.

## 13.5.2 Interconnect Interfaces

The following interconnect interfaces are described in Section 13.7, "Interconnects":

- LED
- Software-Defined Pins (SDP)
- Shared SPI Flash
- MDIO
- $I^2C$
- SMBus
- NC-SI
- Memory-mapped I/O (MMIO)

# 13.6 Basic Operation

## 13.6.1 Transmit (Tx) Data Flow

Tx data flow provides a high-level description of all data/control transformation steps needed for sending Ethernet packets over the wire.

**Table 13-13. Tx Data Flow**

| Step | Description |
|---|---|
| 1 | The host creates a descriptor ring and configures one of the LAN controllers transmit queues with the address location, length, head, and tail pointers of the ring (one of 128 available Tx queues). |
| 2 | The host is requested by the TCP/IP stack to transmit a packet, it gets the packet data within one or more data buffers. |
| 3 | The host initializes the descriptor(s) that point to the data buffer(s) and have additional control parameters that describes the needed hardware functionality. The host places that descriptor in the correct location at the appropriate Tx ring. |
| 4 | The host updates the appropriate Queue Tail Pointer (TDT). |
| 5 | The LAN controllers DMA senses a change of a specific TDT and as a result sends a PCIe request, over the primary integrated I/O interface bus, to fetch the descriptor(s) from host memory. |
| 6 | The descriptor(s) content is received in a PCIe read completion, over the primary integrated I/O interface bus, and is written to the appropriate location in the descriptor queue. |
| 7 | The DMA fetches the next descriptor and processes its content. As a result, the DMA sends PCIe requests, over the primary integrated I/O interface bus, to fetch the packet data from system memory. |
| 8 | The packet data is being received from PCIe completions, over the primary integrated I/O interface bus, and passes through the transmit DMA that performs all programmed data manipulations (various CPU offloading tasks as checksum offload, TSO offload, etc.) on the packet data on the fly. |
| 9 | While the packet is passing through the DMA, it is stored into the transmit FIFO.<br>After the entire packet is stored in the transmit FIFO, it is then forwarded to transmit switch module. |
| 10 | The transmit switch arbitrates between host and management packets and eventually forwards the packet to the MAC |
| 11 | The MAC appends the L2 CRC to the packet and delivers the packet to the integrated PHY. |
| 12 | The PHY performs the PCS encoding, scrambling and the other manipulations required to deliver the packet over the physical interface at the selected speed. |
| 13 | When all the PCIe completions for a given packet are complete, the DMA updates the appropriate descriptor(s). |
| 14 | The descriptors are written back to host memory, over the primary integrated I/O interface bus, using PCIe posted writes. The head pointer is updated in host memory as well. |
| 15 | An interrupt is generated to notify the host driver that the specific packet has been read to the LAN controllers and the driver can then release the buffer(s). |

### 13.6.2 Receive (Rx) Data Flow

Rx data flow provides a high-level description of all data/control transformation steps needed for receiving Ethernet packets.

**Table 13-14. Rx Data Flow**

| Step | Description |
|---|---|
| 1 | The host creates a descriptor ring and configures one of the LAN controllers receive queues with the address location, length, head, and tail pointers of the ring (one of 128 available Rx queues). |
| 2 | The host initializes descriptor(s) that point to empty data buffer(s). The host places these descriptor(s) in the correct location at the appropriate Rx ring. |
| 3 | The host updates the appropriate Queue Tail Pointer (RDT). |
| 4 | A packet enters the PHY through the physical interface. |
| 5 | The PHY performs the required manipulations on the incoming signal de-scrambling, PCS decoding, etc. |
| 6 | The PHY delivers the packet to the Rx MAC. |
| 7 | The MAC forwards the packet to the Rx filter. |
| 8 | If the packet matches the pre-programmed criteria of the Rx filtering, it is forwarded to an Rx FIFO. |
| 9 | The receive DMA fetches the next descriptor from the appropriate host memory ring to be used for the next received packet. |
| 10 | After the entire packet is placed into an Rx FIFO, the receive DMA posts the packet data to the location indicated by the descriptor through the primary integrated I/O interface bus.<br>If the packet size is greater than the buffer size, more descriptor(s) are fetched and their buffers are used for the received packet. |
| 11 | When the packet is placed into host memory, the receive DMA updates all the descriptor(s) that were used by the packet data. |
| 12 | The receive DMA writes back the descriptor content along with status bits that indicate the packet information including what offloads were done on that packet. |
| 13 | The LAN controllers initiates an interrupt to the host to indicate that a new received packet is ready in host memory. |
| 14 | The host reads the packet data and sends it to the TCP/IP stack for further processing. The host releases the associated buffer(s) and descriptor(s) once they are no longer in use. |

# 13.7 Interconnects

## 13.7.1 Setting Up PCI Device Presence and Non-presence

PCI device and function presence or non-presence must be established before PCI enumeration and before the LAN controllers are released from PCI reset. Soft straps for the LAN controllers are provided for this purpose.

### 13.7.1.1 LAN Controllers Soft Straps

There are six soft straps that affect the four LAN ports. By default, all LAN ports and PCI functions of a particular product SKU are enabled. The six soft straps related to Ethernet ports/functions are:

1. LAN Controller 0 – Power in S5 enable.
2. LAN Controller 1 – Power in S5 enable.
3. LAN Controller 0 – Disable.
4. LAN Controller 1 – Disable.
5. LAN Controller 0, Port 1 Disable.
6. LAN Controller 1, Port 1 Disable.

*Note:* LAN0 Port 0 and LAN1 Port 0 cannot be disabled.

The definition of these soft straps can be found in Table 4-3, "Flash Descriptor Soft Straps". These soft strap settings are defined in the flash descriptor (region 0) of the SPI.

## 13.7.2 Disabling PCI Functions by BIOS

The BIOS can disable the LAN controller PCI functions. The memory mapped Function Disable (FUNC_DIS) register of the Power Management Controller (PMC) portion of the Platform Control Cluster (PCC) contains four bits to disable for PCI functions in LAN0 (ports 0 and 1) and LAN1 (ports 0 and 1).

*Note:* LAN port 0 must be enabled if a LAN controller is enabled.

## 13.7.3 LAN Port Interface

The LAN ports, each with its own MAC and set of transmit and receive queues, performs all of the functions required for transmission, reception, and collision handling called out in the standards.

The default link mode is set via the 2-bit *Link Mode* field (bits [5:4]) of the Initialization Control 3 word, located at 16-bit word offset 0x24 of the particular LAN-port base address of the shared SPI Flash. The link modes are listed in Table 13-15, "Link Modes".

**Table 13-15. Link Modes**

| Link Mode | Device ID | Branding String |
|---|---|---|
| Default (10Gbps SKU) | 0x1306 | Intel® Ethernet Connection X553 |
| Default (1Gbps SKU) | 0x1307 | Intel® Ethernet Connection X553 |
| VF (Hyper-V) | 0x15B4 | Intel® X553 Virtual Function |
| KR/KX to Backplane | 0x15C2 [1] | Intel® Ethernet Connection X553 Backplane |
| KX (2.5GbE) | 0x15C3 [2], 0x15C2 [1] | Intel® Ethernet Connection X553 Backplane |
| SFI (Native) [3] | 0x15C4 | Intel® Ethernet Connection X553 10 GbE SFP+ |
| VF | 0x15C5 | Intel® X553 Virtual Function |
| SGMII Backplane | 0x15C6 [1], 0x15C7 [2] | Intel® Ethernet Connection X553 1GbE |
| SGMII | 0x15E4 [1], 0x15E5 [2] | Intel® Ethernet Connection X553 1GbE |
| KR/KX to 10GBASE-T, 1GBASE-T (X557) [3] | 0x15C8 | Intel® Ethernet Connection X553 10GBASE-T |
| KR to SFP+ (InPhi) [3] | 0x15CE | Intel® Ethernet Connection X553 10 GbE SFP+ |

***Notes:***
1. This device ID represents a 10G LAN controller SKU for the particular link mode.
2. This device ID represents a non-10G LAN controller SKU for the particular link mode.
3. Supported only on 10G capable LAN controller SKU.

LAN port features:

- Device power management and power states
- DMA coalescing
- Broadcast Wake Up
- IPv4/IPv6 packet-detect support
- Magic Packet detection and operation
- Packet pattern flexible filters

### 13.7.4 Reference Clock Input

The LAN controllers require a differential, non spread-spectrum reference clock for most of its MAC operation. The source of the clock is the 25-MHz crystal attached to pins CLK_X1_PAD and CLK_X2_PAD of the SoC. See Chapter 7, "Clock Architecture". Routing of this clock signal from the 25-MHz crystal to pins is critical and the Product Design Guide should be followed very carefully.

### 13.7.5 Pin Straps

There are no pin straps associated with the integrated LAN controllers in this SoC generation.

## 13.8 Shared SPI Flash for Use with LAN0 and LAN1

Unlike previous SoC generations, there is not a dedicated external NVM device that stores the Ethernet configuration parameters. LAN0 and LAN1 use two sections of the shared system SPI Flash to configure parameters for all LAN ports and PCI functions including MAC addresses, LED behaviors, receive packet filters for manageability, wake-up capability, etc.

The contents of the shared SPI Flash are always addressed as 16-bit words. When an offset value is shown, it is always in terms of 16-bit words.

Some of the shared SPI Flash 16-bit words are used to specify hardware parameters that affect all four LAN interfaces such as those affecting circuit behavior. Other words are associated with a specific LAN port. Both LAN0 and LAN1 interfaces access the shared SPI Flash to obtain their respective configuration settings.

### 13.8.1 Shared SPI Flash Starter Images

Intel provides a number of NVM flash images for LAN functionality that SoC customers can use with their designs. Please refer to the NVM flash image documentation with the images regarding the descriptions and use of these NVM images. These files are considered starter images and to some extent can be altered for a customer particular usage. Intel provides tools for customers to program images to the system for various operating systems. Contact your local Intel sales representative for details. If Intel needs to provide updates to customers, it provides the necessary software tools to do this.

### 13.8.2 Shared SPI Flash Map Related to LAN0 and LAN1

The shared SPI Flash related to LAN0 and LAN1 is divided into five major regions and a number of sub-regions as listed in Section 4.0 of the “Programmer’s Reference Manual” show a list of words of customer interest. Shared SPI Flash words defined for exclusive use by Intel are not listed.

### 13.8.3 Unique MAC Address

The SoC customer assigns their own unique six-byte MAC address for each LAN port. The shared SPI Flash has three 16-bit word locations defined for each of the four LAN ports. The six-byte MAC address is programmed to the shared SPI Flash as shown in this example, where the desired MAC address for LAN port 2 is 00-A0-C9-00-00-03:

- Word 0xC0 = 0xA000
- Word 0xC1 = 0x00C9
- Word 0xC2 = 0x0300

### 13.8.4 VLAN Support

Two basic types of VLANs are supported:

- Tagged VLANs are based on the IEEE 802.1Q specification. Each packet has a 4-byte tag added to the packet header. The switch must support IEEE 802.1Q tagging and be properly configured. Check the switch documentation for the correct switch configuration.
- Untagged or port-based VLANs are statically configured on the switch and are transparent to connected devices.

## 13.9 MMIO and Software Interface

The MMIO register descriptions for each of the PCI functions of the LAN controllers are not described in this document.

## 13.10 Host Primary Interface

The LAN controllers communicate with the host CPUs and host memories over an internal fabric.

### 13.10.1 Host Interface Architecture, Transaction and Link Layer Properties

- All PCI functions are native PCIe functions
- Credit-based flow control
- Packet sizes/formats:
  - Maximum upstream (write) payload size of 512 bytes
    Configurable sizes are: 128 bytes through 512 bytes
  - Maximum downstream (read) payload size of 2 KB
    Configurable sizes are: 128 bytes through 2 KB
- Reset/initialization:
  - Credit negotiation performed by hardware
- Transaction layer mechanisms
  - 64-bit memory address spaces
  - Removal of I/O BAR (optional)
  - Relaxed ordering
  - Flow control update time out mechanism
  - ID-based ordering (IDO)
  - Function-Level Reset (FLR)
  - Reliability
  - Advanced Error Reporting (AER)
- Power management
  - Wake capability
  - Latency Tolerance Reporting (LTR)
- The LAN controllers support the following extended capabilities:
  - Device serial number
  - Alternative Routing-ID Interpretation (ARI)
  - Single Root I/O Virtualization (SR-IOV)
  - Access Control Services (ACS)
- Software configuration mechanism:
  - Uses PCI configuration and bus enumeration model
  - PCIe-specific configuration registers mapped via PCI extended capability mechanism

## 13.10.2 PCIe Transaction Layer

### 13.10.2.1 Transaction Types Accepted by the LAN Controllers

Table 13-16, "Transaction Types Accepted by the Transaction Layer" lists the transactions accepted by the device and their attributes.

**Table 13-16. Transaction Types Accepted by the Transaction Layer**

| Transaction Type | Tx Layer Reaction | Hardware Should Keep Data From Original Packet |
|---|---|---|
| Configuration Read Request | CPLH + CPLD | Requester ID, TAG, attribute |
| Configuration Write Request | CPLH | Requester ID, TAG, attribute |
| Memory Read Request | CPLH + CPLD | Requester ID, TAG, attribute |
| Memory Write Request | - | - |
| I/O Read Request | CPLH + CPLD | Requester ID, TAG, attribute |
| I/O Write Request | CPLH | Requester ID, TAG, attribute |
| Read Completions | - | - |
| Message | - | - |

#### 13.10.2.1.1 Size of Target Accesses

##### Memory Accesses

Rules for accesses to the CSR space (both memory BAR and MSI-X BAR):

- Write accesses
  - Zero-length writes have no internal impact (nothing written, no effect such as clear-by-write). The transaction is treated as a successful operation (no error event).
- CSR writes are 32-bit or 64-bit only. Larger or partial CSR writes are handled as completer abort - data is dropped and an error is generated per PCIe rules read accesses
  - Partial reads with at least one byte disabled are handled as a full read. Any side effect of the full read (such as clear by read) is also applicable to partial reads. The completion on PCIe follows the specification rules regarding the number of bytes reported in the completion.
  - Zero-length reads generate a completion, but the register is not accessed and undefined data is returned.
  - CSR reads are 32-bit or 64-bit only. Larger CSR read requests are handled as completer abort - the completion includes a CA status and an error is generated per PCIe rules.
  - Some 64-bit reads are handled atomically (such as not interleaved with any other requests). This applies mainly to reading counters, where all 64 bits need to be read simultaneously. Such registers are explicitly marked in their description.

Rules for accessing the shared SPI Flash space in the memory BAR:

- Write accesses
  - Writes to the shared SPI Flash are 16-bit wide only
  - Any larger write accesses are handled as completer abort - data is dropped and an error is generated per PCIe rules
- Read accesses
  - Reads to shared SPI Flash are 16-bit wide
  - Partial reads with at least one byte disabled are handled internally as a full read. That is, any side effect of the full read (such as clear by read) is also applicable to partial reads. The completion on PCIe follows the specification rules regarding the number of bytes reported in the completion.
  - Larger CSR read requests are handled as completer abort; the completion includes a CA status and an error is generated per PCIe rules

#### I/O Accesses

Rules for accesses to the I/O BAR:

- Write accesses
  - Write accesses are 32-bit wide
  - Zero-length writes have no internal impact (nothing written, no effect such as clear-by-write). The transaction is treated as a successful operation (no error event).
  - Other accesses (partial writes, larger writes) are handled as completer abort - data is dropped and an error is generated per PCIe rules
- Read accesses
  - Reads to the I/O BAR are 32-bit wide
  - Partial reads with at least one byte disabled are handled internally as a full read. That is, any side effect of the full read (such as clear by read) is also applicable to partial reads. The completion on PCIe follows the specification rules regarding the number of bytes reported in the completion.
  - Larger CSR read requests are handled as completer abort - the completion includes a CA status and an error is generated per PCIe rules

### 13.10.2.1.2 Support for Dynamic Changes

The LAN controllers capture the bus number and device number per each configuration write request. However, a dynamic change of the bus number or device number is not supported. Rather, the PCIe link should be quiescent prior to such a change, including reception of all completion for previous requests.

### 13.10.2.2 Transaction Types Initiated by the LAN Controllers

**Table 13-17. Transaction Types Initiated by the Transaction Layer**

| Transaction Type | Payload Size | Tx Layer Reaction |
|---|---|---|
| Configuration Read Request Completion | Dword | CPLH + CPLD |
| Configuration Write Request Completion | - | CPLH |
| I/O Read Request Completion | Dword | CPLH + CPLD |
| I/O Write Request Completion | - | CPLH |
| Read Request Completion | Dword/Qword | CPLH + CPLD |
| Memory Read Request | - | NPH |
| Memory Write Request | <= MAX_PAYLOAD_SIZE | PH + PD |
| Message | 64 bytes | PH |

Configuration values:

- Max Payload Size - The value of the Max_Payload_Size Supported field in the Device Capabilities register is loaded from the shared SPI Flash.
  - Hardware default is 512 bytes.
  - System software then programs the actual value into the Max_Payload_Size field of the Device Control register.

    Non-ARI mode: If not all functions are programmed with the same value, the max payload size used for all functions is the minimum value programmed among all functions.

    ARI mode: Max_Payload_Size is determined solely by the setting in Function 0
- Max_Read_Request_Size - The SoC LAN controllers support read requests of up to 2 KB.
  - The actual maximum size of a read request is defined as the minimum (2 KB, Max_Read_Request_Size field in the Device Control register).

The number of outstanding memory read requests is bounded by the following:

- The total number of outstanding requests is not more than 32 requests. These are shared by all sources for memory reads.

#### 13.10.2.2.1 Data Alignment

Requests must never specify an address/length combination that causes a memory space access to cross a 4 KB boundary. The LAN controllers therefore break requests into 4 KB-aligned requests (if needed). This does not place any requirement on software. However, if software allocates a buffer across a 4 KB boundary, hardware issues multiple requests for the buffer. Software should consider aligning buffers to a 4 KB boundary in cases where it improves performance. The maximum size of a read request is defined as the minimum (2 KB bytes, Max_Read_Request_Size).

The general rules for packet alignment are as follows. Note that these apply to all LAN controller requests (read/write):

- The length of a single request does not exceed the PCIe limit of MAX_PAYLOAD_SIZE(512B) for write and MAX_READ_REQ (2KB) for read.
- The length of a single request does not exceed the LAN controller internal limitations.
- A single request does not span across different memory pages as noted by the 4 KB boundary alignment previously mentioned.

If a request can be sent as a single packet and still meet the general rules for packet alignment, then it is not broken at the cache line boundary but rather sent as a single packet. However, if any of the three general rules require that the request is broken into two or more packets, then the request is broken at the cache line boundary.

For requests with data payload, if the payload size is larger than (MAX_PAYLOAD_SIZE - CACHELINE_SIZE), then the request is broken into multiple TLPs starting at the first cache line boundary following the (MAX_PAYLOAD_SIZE - CACHELINE_SIZE) bytes. For example, if MAX_PAYLOAD_SIZE = 256 bytes and CACHELINE_SIZE = 64 bytes, a 1 KB request starting at address 0x...10 is broken into TLPs such that the first TLP contains 240bytes of payload (since 240 bytes + 0x10 = 256 bytes is on cache line boundary).

The system cache line size is controlled by the *PCI_CNF2.CACHELINE_SIZE* bit, loaded from the shared SPI Flash update.

*Note:* The Cache Line Size register in the PCI configuration space is not related to the *PCI_CNF2.CACHELINE_SIZE* and is solely for software use.

### 13.10.2.3 Messages

#### 13.10.2.3.1 Received Messages

Message packets are special packets that carry a message code. The upstream device transmits special messages to the LAN controllers by using this mechanism. The transaction layer decodes the message code and responds to the message accordingly.

**Table 13-18. Supported Message in the LAN Controllers (as a Receiver)**

| Message Code [7:0] | Routing | Message | LAN Controller Later Response |
|---|---|---|---|
| 0x40, 0x41, 0x43, 0x44, 0x45, 0x47, 0x48 | 100b | Ignored messages (used to be hot-plug messages) | Silently drop. |
| 0x50 | 100b | Slot power limit support (has one Dword data) | Silently drop. |
| 0x7E | 000b, 010b, 011b, 100b | Vendor defined type 0 | Drop and handle as an Unsupported request |
| 0x7F | 100b | Vendor defined type 1 | Silently drop. |
| 0x7F | 010b, 011b, 000b | Vendor defined type 1 | Silently drop. |
| 0x00 | 011b | Unlock | Silently drop. |

#### 13.10.2.3.2 Transmitted Messages

The transaction layer is also responsible for transmitting specific messages to report internal/external events (such as interrupts and PMEs).

### 13.10.2.4 Transaction Attributes

#### 13.10.2.4.1 Traffic Class (TC) and Virtual Channels (VC)

The LAN controllers only supports TC = 0b and VC = 0b (default).

### 13.10.2.5 Ordering Rules

The LAN controllers meet the PCIe ordering rules by following the PCI simple device model:

1. Deadlock Avoidance – The LAN controllers meets the PCIe ordering rules that prevent deadlocks:
   a. Posted writes overtake stalled read requests. This applies to both target and master directions. For example, if master read requests are stalled due to lack of credits, master posted writes are allowed to proceed. On the target side, it is acceptable to timeout on stalled read requests in order to allow later posted writes to proceed.
   b. Target posted writes overtake stalled target configuration writes.
   c. Completions overtake stalled read requests. This applies to both target and master directions. For example, if master read requests are stalled due to lack of credits, completions generated by the LAN controllers are allowed to proceed.
2. Descriptor/Data Ordering — The LAN controllers ensure that a Rx descriptor is written back on PCIe only after the data that the descriptor relates to is written to the PCIe link.
3. MSI and MSI-X Ordering Rules — System software might change the MSI or MSI-X tables during run-time. Software expects that interrupt messages issued after the table has been updated are using the updated contents of the tables.
   a. Since software doesn't know when the tables are actually updated in the LAN controllers, a common scheme is to issue a read request to the MSI or MSI-X table (a PCI configuration read for MSI and a memory read for MSI-X). Software expects that any message issued following the completion of the read request, is using the updated contents of the tables.
   b. Once an MSI or MSI-X message is issued using the updated contents of the interrupt tables, any consecutive MSI or MSI-X message does not use the contents of the tables prior to the change.
4. The LAN controllers meet the rules relating to independence between target and master accesses:
   a. The acceptance of a target posted request does not depend upon the transmission of any TLP.
   b. The acceptance of a target non-posted request does not depend upon the transmission of a non-posted request.
   c. Accepting a completion does not depend upon the transmission of any TLP.

#### 13.10.2.5.1 Relaxed Ordering

The LAN controllers take advantage of the relaxed ordering rules in PCIe. By setting the relaxed ordering bit in the packet header, the LAN controllers enable the system to optimize performance in the following cases:

1. Relaxed ordering for descriptor and data reads — When the LAN controller masters a read transaction, its split completion has no ordering relationship with the writes from the CPUs (same direction). It should be allowed to bypass the writes from the CPUs.
2. Relaxed ordering for receiving data writes — When the LAN controllers master receive data writes, it also enables them to bypass each other in the path to system memory because software does not process this data until their associated descriptor writes are done.
3. The LAN controllers cannot relax ordering for receive descriptor writes or an MSI write.

Relaxed ordering is enabled in the LAN controllers by clearing the *CTRL_EXT.RO_DIS* bit. Relaxed ordering is further controlled through the *Enable Relaxed Ordering* bit in the PCIe Device Control register.

#### 13.10.2.5.2 ID-based Ordering (IDO)

IDO was introduced in the PCIe rev. 2.1 specification. When enabled, The LAN controllers set IDO in all applicable TLPs defined in the PCIe specification.

This capability enables a supporting root complex to relax ordering rules for TLPs sent by different requesters.

IDO is enabled when all of the following conditions are met:

- The shared SPI Flash update *PCI_CAPSUP.IDO* Enable bit is set
- The PCIe *IDO Request Enable* bit (for requests) or the *IDO Completion Enable* bit (for completions) in Device Control 2 register is set

## 13.10.3 Error Events and Error Reporting

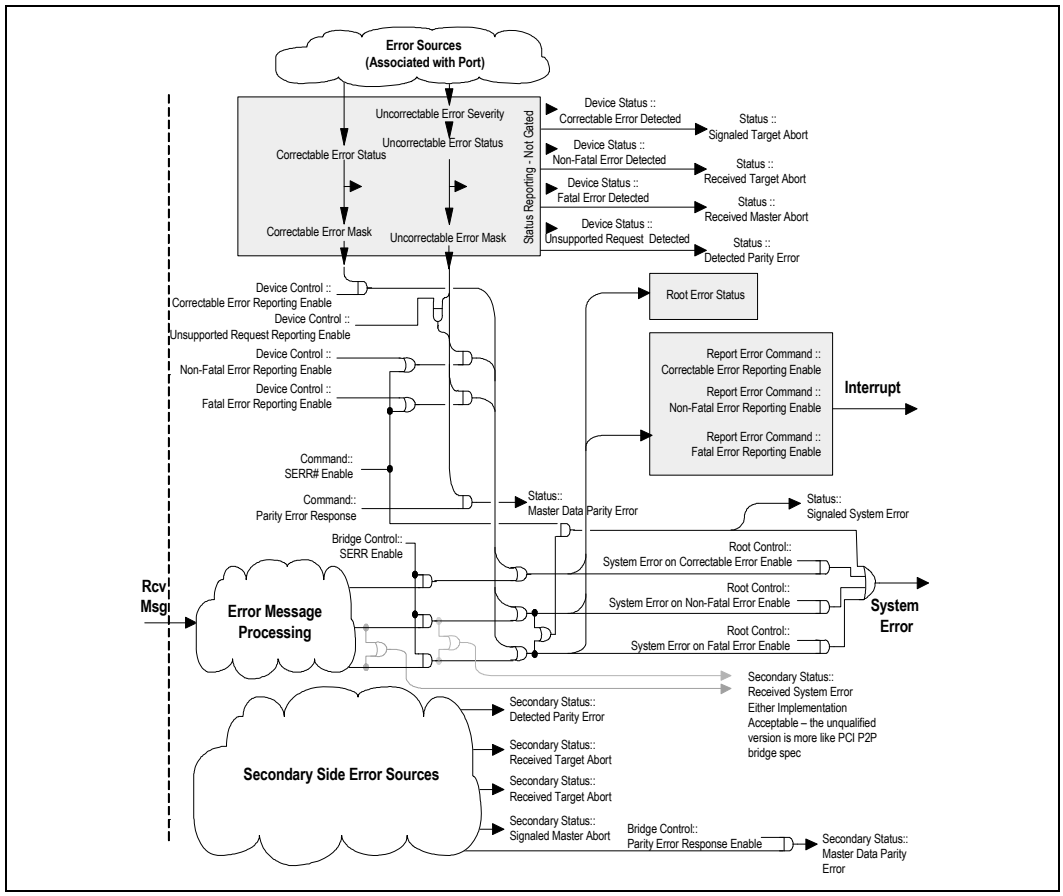
### 13.10.3.1 General Description

PCIe defines three error reporting paradigms: the baseline capability, the Advanced Error Reporting (AER) capability, and a proprietary mechanism. The baseline error reporting capabilities are required of all PCIe devices and define the minimum error reporting requirements. The AER capability is defined for more robust error reporting and is implemented with a specific PCIe capability structure. Both mechanisms are supported by the LAN controllers. The proprietary error reporting mechanism used for error is better handled by the software device driver using internal CSRs.

The *SERR# Enable* and the *Parity Error* bits from the Legacy Command register also take part in the error reporting and logging mechanism.

In a multi-function device, PCIe errors that are not related to any specific function within the device are logged in the corresponding status and logging registers of all functions in that device. Figure 13-3, "Error Reporting Mechanism" shows, in detail, the flow of error reporting in the LAN controllers.

**Figure 13-3. Error Reporting Mechanism**



### 13.10.3.2 Error Events

Table 13-19, "Response and Reporting of PCIe Error Events" lists the error events identified by the LAN controller and the response in terms of logging, reporting, and actions taken. Refer to the *PCI Express Base Specification Revision 3.0* for the effect on the PCI Status register.

**Table 13-19. Response and Reporting of PCIe Error Events**

| Error Name | Error Events | Default Severity | Action |
|---|---|---|---|
| TLP Errors | | | |
| Poisoned TLP Received | • TLP With Error Forwarding (EP = 1b)<br>• Data Parity | Uncorrectable ERR_NONFATAL Log Header | If completion TLP:<br>Error is non-fatal (default case)<br>• Send error message if advisory<br>• Retry the request once and send advisory error message on each failure<br>• If fails, send uncorrectable error message<br>Error is defined as fatal<br>• Send uncorrectable error message |
| Unsupported Request (UR) | • Receipt of TLP with unsupported Request Type<br>• Receipt of an Unsupported Vendor Defined Type 0 Message<br>• Invalid Message Code<br>• Wrong Function Number<br>• Received TLP Outside BAR Address Range<br>• Receipt of a Request TLP during D3hot, other than Configuration and Message requests | Uncorrectable ERR_NONFATAL Log header | Send Completion With UR |
| Completion Timeout | Completion Timeout Timer Expired | Uncorrectable ERR_NONFATAL | Error is non-fatal (default case)<br>• Send error message if advisory<br>• Error is defined as fatal<br>• Send uncorrectable error message |
| Completer Abort | Received Target Access With illegal data size per Section 13.10.2.1.1, "Size of Target Accesses" | Uncorrectable. ERR_NONFATAL Log header | Send completion with CA |
| Unexpected Completion | Received Completion Without a Request For It (Tag, ID, etc.) | Uncorrectable ERR_NONFATAL Log Header | Discard TLP |
| Malformed TLP (MP) | • Data Payload Exceed Max_Payload_Size Received<br>• TD field value does not correspond with the observed size<br>• PM Messages That Don't Use TC0<br>• Usage of Unsupported VC<br>• Target request crosses a 4KB boundary | Uncorrectable ERR_FATAL Log Header | Drop the Packet, Free FC Credits |
| Completion with Unsuccessful Completion Status | | No Action (already done by originator of completion) | Free FC Credits |
| Command Parity | Parity error is detected on the command | Uncorrectable ERR FATAL | |

### 13.10.3.3 Completion Timeout Mechanism

- The LAN controllers support completion time out as defined in the PCIe specification.
- The LAN controllers control the following aspects of completion time out:
- Disabling or enabling completion timeout
  - The PCIe *Completion Timeout Disable Supported* bit in the Device Capabilities 2 register is hard wired to 1b to indicate that disabling completion timeout is supported
  - The PCIe *Completion Timeout Disable* bit in Device Control 2 register controls whether completion timeout is enabled
- A programmable range of timeout values
  - The LAN controllers support all four ranges as programmed in the *Completion Time out Ranges Supported* field of the Device Capabilities 2 register. The actual completion time out value is written in the *Completion Time out Value* field of Device Control 2 register.

The following sequence takes place when completion timeout is detected:

- The appropriate message is sent on PCIe as listed in Table 13-19, "Response and Reporting of PCIe Error Events"
- The affected queue or client takes action based on the nature of the original request.
- An interrupt is issued to the respective PF.

### 13.10.3.4 Error Forwarding (TLP Poisoning)

If a TLP is received with an error-forwarding trailer, the packet is dropped and is not delivered to its destination, the LAN controller then reacts as listed in Table 13-19.

The following sequence takes place when a poisoned TLP is received:

- The appropriate message is sent on PCIe as listed in Table 13-19, "Response and Reporting of PCIe Error Events".
- An interrupt is issued.
- If the TLP is a completion, a completion time out follows at some later time. Processing continues as described in Section 13.10.2.3, "Messages".

System logic is expected to trigger a system-level interrupt to signal the operating system of the problem. Operating systems can then stop the process associated with the transaction, re-allocate memory to a different area instead of the faulty area, etc.

### 13.10.3.5 Completion With Unsuccessful Completion Status

A completion arriving with an unsuccessful completion status (either UR or CA) is dropped and not delivered to its destination. A completion time out follows at some later time. Processing continues as described in Section 13.10.2.3, "Messages".

### 13.10.3.6 Blocking on Upper Address

The PCI_UPADD register blocks master accesses from being sent out on PCIe if the TLP address exceeds some upper limit. Bits [31:1] correspond to bits [63:33] in the PCIe address space, respectively.

When a bit is set in GLPCI_UPADD[31:1], any transaction, in which the corresponding bit in its address is set, is blocked and not sent over PCIe. If all register bits are cleared, there is no effect (such as no TLPs are blocked by this mechanism).

The PCI_UPADD register is loaded from the shared SPI Flash update with a value allowing all addresses to pass. The software device driver should override this value with a system dependent value.

Processing a blocked transaction:

  — Write transaction:

  — The transaction is dropped.

  — Set the Exceeded upper address limit (write requests) event in the PCIe errors register (see Section 13.10.3.7).

  — An interrupt is issued as described in Section 13.10.3.7.

- Read transaction:

  — The transaction is dropped.

  — Set the Exceeded upper address limit (read requests) event in the PCIe errors register (see Section 13.10.3.7).

  — The originating internal client is notified.

  — The affected queue or client takes action based on the nature of the original request. An interrupt is issued to the respective PF.

### 13.10.3.7 Proprietary Error Reporting

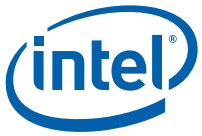
The PCIe specification defines how to report errors to system software. There are, however, error events that the software device driver should be aware of or that the software device driver is in better position to handle and recover from. This section describes the mechanism to report PCIe related errors to software device drivers.

Several CSRs are dedicated to this functionality, with a separate bit allocated per error type (see Table 13-20):

- The PCIe Errors Reported register (PCI_PCIERR - RO) indicates which errors are reported using this mechanism. It is shared by all PFs. It is loaded from the shared SPI Flash update. All non-reserved errors are enabled.
- The PCIe Interrupt Cause register (PCI_ICAUSE - RW1C) indicates pending errors for errors set in the PCIe Errors Reported register. It is dedicated per PF.
- The PCIe Interrupt Enable register (PCI_IENA - RW) determines if an interrupt should be issued to the respective PCI function on an error event. It is dedicated per PF.

Reporting an error to the PF driver involves the following steps:

- The Auto-Negotiate (AN) controllers check if the respective bit is set in the PCIe Errors Reported register. If cleared, done. Else, continue.
- The respective bit is set in the PCIe Interrupt Cause register.
- If the respective bit is set in the PCIe Interrupt Enable register, an interrupt is issued to the PCI function. The PCI_EXCEPTION cause is used (see the EICR register described in the PRM).



### Table 13-20. PCIe Errors Reported to Device Software

| Error Event | Index | Description and Comments | Function Association |
|---|---|---|---|
| Exceeded upper address limit (read requests) | 00 | See Section 13.10.3.6, "Blocking on Upper Address" | Sent to PF |
| Exceeded upper address limit (write requests) | 01 | See Section 13.10.3.6, "Blocking on Upper Address" | Sent to PF |
| Reserved | 02 | Reserved entries | N/A |
| Poisoned TLP received | 03 | See Section 13.10.3.4, "Error Forwarding (TLP Poisoning)" | Sent to PF |
| Reserved | 04-05 | Reserved entries | N/A |
| Unsupported Request - Request Type | 07 | Request causes an Unsupported Request due to receipt of TLP with unsupported Request Type | Sent to PF |
| Unsupported Request - Vendor Message | 08 | Request causes an Unsupported Request due to receipt of an Unsupported Vendor Defined Type 0 Message | Sent to PF unless r[2:0] = *Broadcast from Root Complex,* in which case sent to all PFs |
| Reserved | 09 | Reserved entries | N/A |
| Unsupported Request - Function Number | 10 | Request causes an Unsupported Request due to receipt of a not-supported Function Number | Sent to all PFs |
| Reserved | 12 - 11 | Reserved entries | N/A |
| Completer abort - target size | 13 | Received Target Access with illegal data size per Section 13.10.2.1.1, "Size of Target Accesses" (CA) | Sent to PF |
| Reserved | 14 - 31 | Reserved entries | N/A |

## 13.11 Management Interfaces

The LAN controllers contain three possible interfaces to an external BMC.

- SMBus
- NC-SI (over RMII)
- MCTP (over SMBus)

LAN0 and LAN1 can support SMBus and NCSI independently. This is done through configuration in the shared NVM section for LAN0 and LAN1. Sideband management is typically off by default.

### 13.11.1 SMBus

SMBus is an optional interface for pass-through and/or configuration traffic between an external BMC and the LAN controllers. The SMBus channel behavior and the commands used to configure or read status from the LAN controllers are described in the PRM.

The LAN controllers also enable reporting and controlling the device using the MCTP protocol over SMBus. The MCTP interface is used by the BMC to control the port and for pass-through traffic. All network ports are mapped to a single MCTP endpoint on SMBus.

#### 13.11.1.1 Channel Behavior

The SMBus specification defines a maximum frequency of 100 KHz. However, when acting as a slave, the LAN controllers can receive transactions with a clock running at up to 1 MHz. When acting as a master, they can toggle the clock at 100 KHz, 400 KHz or 1 MHz. The speed used is set by the *SMBus Connection Speed* field in the SMBus Notification Time out and Flags shared SPI Flash word as described in the PRM.

### 13.11.2 NC-SI Transactions

The NC-SI link supports both pass-through traffic between the BMC and the LAN controller LAN functions, as well as configuration traffic between the BMC and the SoC LAN controller internal units as defined in the NC-SI protocol.

### 13.11.3 MCTP (over SMBus)

The LAN controllers support MCTP protocol for management. MCTP runs over the SMBus. The LAN controller also implements MCTP over NC-SI for command and pass-through traffic.

## 13.12 Sideband Interface (NC-SI)

The NC-SI interface in the LAN controllers is a connection to an external MC. It operates as a single interface with an external BMC, where all traffic between the LAN controllers and the BMC flow through the interface.

The LAN controller NC-SI interface meets the NC-SI version 1.0.0 specification as a PHY-side device.

### 13.12.1 Electrical Characteristics

The LAN controllers comply with the electrical characteristics defined in the NC-SI specification.

## 13.13 Shared SPI Flash

### 13.13.1 General Overview

Enablement, configuration and addressing are configured in the sections of the system shared SPI Flash specific to LAN0 and LAN1. All management interfaces related to LAN0 and LAN1 are disabled by default and they must be enabled via the shared SPI Flash.

The LAN controllers (LAN0 and LAN1) use sections of the system shared SPI Flash specific to LAN0 and LAN1 to enable, configure and personalize each port. The shared SPI Flash section is divided into a few general regions:

- Hardware Accessed — Loaded by the LAN controller hardware after power up, PCI reset deassertion, D3 to D0 transition, or software reset. Different hardware sections in the shared SPI Flash are loaded at different events.
- Firmware Area — Includes firmware code and structures used by the firmware for management configuration in its different modes.
- Software Accessed — This region is used by software entities such as LAN drivers, option ROM software and tools, PCIe bus drivers, VPD software, etc.

### 13.13.2 Shared SPI Flash Protection

The contents of several shared SPI Flash update modules must be protected via authentication.

The shared SPI Flash protection method implemented in the LAN controllers rely on an authenticate on update concept. It means that protected modules are not authenticated after initialization, but prior to committing a module update operation only. Shared SPI Flash protection is guaranteed by an inductive authentication chain, that assumes an initial secured shared SPI Flash image, and requires that any shared SPI Flash update must be secure as well. This method mandates the following limitations and restricting working assumptions:

1. An initial good image is loaded into the shared SPI Flash at the manufacturing site, which is assumed to be safe.
   a. It assumes customers (OEM and end-user) know the source of the installed components, the supply chain producing these components is not compromised during manufacturing, and that the SoC is physically protected from modification after deployment.
   b. The possibility exists that unauthorized firmware might be loaded into the shared SPI Flash via physical modification post manufacturing, as well as through supply chain vulnerabilities. However, firmware updates via programmatic (software) methods are enhanced to require authentication prior to updating shared SPI Flash settings. Furthermore, host software can independently detect whether the firmware image has an invalid digital signature.
2. In a normal operating mode, shared SPI Flash write accesses are controlled by the device (firmware) and cannot be directly performed by the host. Memory mapped shared SPI Flash access remains available for shared SPI Flash read accesses only. For simplicity and flexibility reasons, shared SPI Flash write accesses from the host can be initiated via *S*oftware Host Interface commands, VPD write interface, or via a BMC command, which are all handled by firmware. All other direct access write modes are blocked by hardware when the shared SPI Flash is protected.

### 13.13.3 Shadow RAM

The first eight 4 KB sectors of the LAN controller shared SPI Flash are allocated to create two 16 KB sections (section 0 and section 1), for the configuration content. At least one of these two sections must be valid at any given time; otherwise, the SoC LAN controllers are set based on its hardware default. Following a Power On Reset, the LAN controllers copy the valid section of the shared SPI Flash device into an internal shadow RAM. Any further read accesses by software or firmware to the lower 16 KB addresses of the shared SPI Flash (not through a Flash BAR) are directed to the internal shadow RAM. Modifications made to the shadow RAM contents are copied by the LAN controllers to the other 16 KB section of the shared SPI Flash, circularly flipping the valid section between sections 0 and 1 in the shared SPI Flash. This mechanism provides the following advantages:

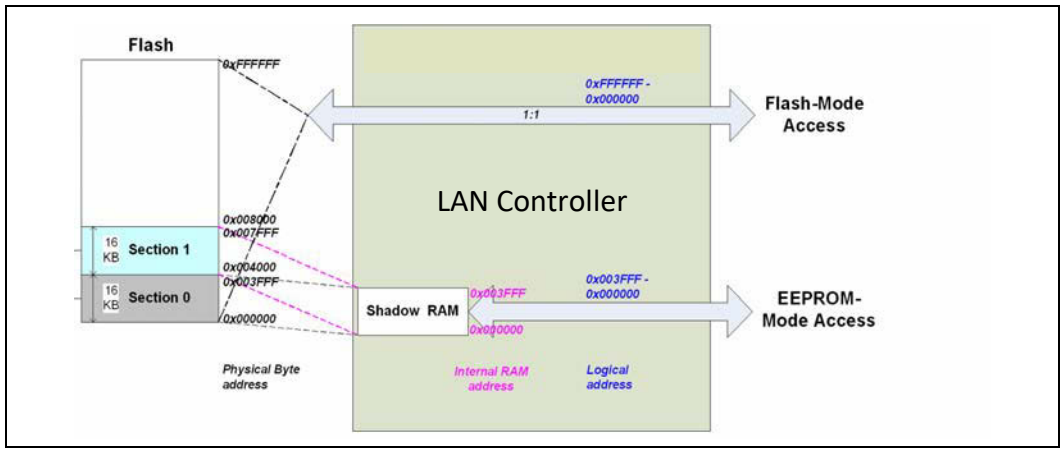
1. A seamless backward-compatible interface for software/firmware to access the first 16 KB of the shared SPI Flash as if an external EEPROM device was connected. This interface is referred as EEPROM-mode access to the shared SPI Flash.

*Note:* Write access to the physical shared SPI Flash is only available in blank Flash/ non-secure modes.

2. A way to protect the image-update procedure from power down events by establishing a double-image policy. See Section 13.13.8.9, "Software Flash Read Flow via the Flash-mode Interface" for a description of the double-image policy. It relies on having pointers to shared SPI Flash modules stored in the shared SPI Flash section mirrored in the internal shadow RAM.
3. A way to ensure hardware auto-load during a PCIe reset event can be completed within PCIe timing constraints (100 ms), even if the shared SPI Flash is busy performing an erase operation initiated prior to that reset event.

Figure 13-4 shows the shadow RAM mapping and interface.

**Figure 13-4. Shared SPI Flash Shadow RAM**



#### 13.13.3.1 Shadow RAM Update Flow

1. Following a write access by the software device driver to update the shadow RAM, the data should be updated in the shared SPI Flash as well. The LAN controllers update the shared SPI Flash from the shadow RAM when software explicitly requests an update using the Shadow RAM Dump Host Interface command. To reduce shared SPI Flash update operations, software is expected to request a dump only once its last shadow RAM write access completes. The LAN controllers then copy the contents of the shadow RAM to the non-valid configuration section and makes it the valid one.
2. The software device driver should wait for the command completion to make sure the flow succeeded. Software should be aware that programming the shared SPI Flash might require a long time since the shared SPI Flash update sequence is handled by firmware. The sector erase command by itself can last hundreds of milliseconds. The regular timeout of the host interface commands should also be sufficient to cover this command also.

## 13.13.4 Shared SPI Flash Clients and Interfaces

There are different software clients that can access the shared SPI flash sections related to LAN0 and LAN1: driver, tools, BIOS, VPD, etc. Table 13-21, "Clients and Access Types to the Shared SPI Flash" lists the different shared SPI Flash access methods.

**Table 13-21. Clients and Access Types to the Shared SPI Flash**

| Client | Shared SPI Flash Access Method (Data Width) | Shared SPI Flash Access Target | Logical Byte Address Range | Shared SPI Flash Access Interface | Protection and Enforcement |
|---|---|---|---|---|---|
| Host Software | Memory BAR (parallel 32-bit) | Flash | 0x000000 - 0xFFFFFF | Memory mapped 32-bit read/write via BARs. | Write access is limited to a single byte and is allowed only if hardware protection is disabled. (FLA.LOCKED = 0b). |
| Host Software | FLA bit banging (serial 1-bit) | Flash | 0x000000 - 0xFFFFFF | Software accesses to shared SPI Flash by toggling the SPI pins. | FLA interface access is available only if hardware protection is disabled (FLA.LOCKED = 0b). |
| Host Software | Host interface Shadow RAM Read/Write command | Shadow RAM | 0x000000 - 0x003FFF | Access to shadow RAM through the Shadow RAM Read/Write command | Requires a valid firmware image.<br>Write protection is enforced by firmware. |
| Host Software | Host interface Shared SPI Flash read command | Flash | 0x000000 - 0xFFFFFF | Software read from the shared SPI Flash via Flash Read command. | Requires a valid firmware image. |
| Host Software | Host interface shared SPI Flash write command/shared SPI Flash sector erase command | Flash | 0x008000 - 0xFFFFFF | Software writes and sector erase. | Requires a valid firmware image.<br>Writes and erases to protected areas are dropped when firmware protection is enabled. |
| Host Software | VPD access (parallel 32-bit) | Shadow RAM | 0x000000 - 0x0003FF | VPD Address and Data registers. | Write accesses are enabled to the R/W area of the VPD.<br>If the VPD structure is not valid, the entire 1024 bytes area becomes RO to VPD accesses. |

### 13.13.5 Memory Mapped Host Interface

The shared SPI Flash is accessed by the LAN controllers each time the host CPU performs a read operation to a memory location mapped to the shared SPI Flash address space, or upon boot via accesses to the space indicated by the Base Address register. Accesses to the shared SPI Flash is based on a direct decode of CPU accesses to a memory window defined in the Memory CSR + Flash Base Address register (PCIe Control register at offset 0x10).

The LAN controllers control accesses to the Flash when it decodes a valid access. Attempts to out of range read access, the PCIe expansion/option ROM module (according to shared SPI Flash size field in Shared SPI Flash Control Word 1) would return a value of 0xDEADBEEF. Attempts to memory-mapped write accesses to the shared SPI Flash when protection is enabled or via expansion ROM BAR are ignored.

## 13.13.6 Shared SPI Flash Access Contention

Shared SPI Flash accesses initiated through different LAN functions might occur concurrently. The LAN controllers do not synchronize between entities accessing the shared SPI Flash, so a contention caused from one entity reading and another modifying the same location is possible.

To avoid such contention between software LANs or between software and firmware accesses, these entities are required to make use of the semaphore registers. Any read or write access to the shared SPI Flash made by software/firmware must be preceded by acquiring ownership over the shared SPI Flash. This is also useful to avoid the time out of a PCIe transaction made to a memory-mapped shared SPI Flash address while the shared SPI Flash is busy performing a sector erase operation.

However, two software entities cannot use this semaphore mechanism: BIOS access through expansion ROM and VPD software.

- Since VPD software accesses only the VPD module, which is located in the configuration section of the shared SPI Flash, VPD accesses are always performed against the shadow RAM. Firmware must take shared SPI Flash ownership before dumping the VPD changes to the shared SPI Flash. The Shadow RAM dump sequence is described in Section 13.13.3.1, "Shadow RAM Update Flow".
- No contention can occur between the BIOS access through expansion ROM and other software entities (including VPD) as it accesses the shared SPI Flash while the operating system is down.
- Contentions between BIOS and firmware can however happen if a system reboot occurs while the MC is accessing the shared SPI Flash.
  - If a system reboot is caused by a user pressing the standby button, it is required to route the wake-up signal from the standby button to the MC and not to the chipset. The MC issues a system reboot signal to the chipset only after the shared SPI Flash write access completes. Firmware is responsible to respond with a busy error code to MC NC-SI commands while other shared SPI Flash update writes are in progress.

If a system reboot is issued by a local user on the host, there is no technical way to prevent shared SPI Flash access contentions between the BIOS and the MC.

***Caution:*** It is the user responsibility when remotely accessing the shared SPI Flash via the MC, to make sure another user is not currently initiating a local host reboot.

*Note:* Other MAC auto-load events are performed from the internal shadow RAM and do not compete with memory mapped accesses to the shared SPI Flash device.

*Note:* Software and firmware should avoid holding shared SPI Flash ownership (via the dedicated semaphore bit) for more than 500 ms.

### 13.13.6.1 Flash Deadlock Avoidance

The Flash is a shared resource among the following clients:

1. Hardware auto-load of shadow RAM (at power up); the LAN controller.
2. LAN port 0 and LAN port 1 software accesses.
3. Manageability/firmware accesses.
4. Software tools.

All clients can access the shared SPI Flash using parallel access. Hardware implements the actual access to the shared SPI Flash. Hardware arbitrates among the different clients and schedules these accesses, avoiding starvation of any client.

However, in the LAN controllers, the software and firmware clients can access the shared SPI Flash using bit banging. In this case, there is a request/grant mechanism that locks the shared SPI Flash to the exclusive use of one client. If one client is stuck without releasing the lock, the other clients can no longer access the shared SPI Flash. To avoid this deadlock, the LAN controllers implement a time-out mechanism, which revokes the grant from a client that doesn't toggle the shared SPI Flash bit-bang interface (*FLA.FL_SCK* bit) for more than two seconds. If any client fails to release the shared SPI Flash interface, hardware clears its grant, enabling the other clients to use the interface.

The deadlock timeout mechanism is enabled by the *Deadlock Timeout Enable* bit in the Control Word 2 in the shared SPI Flash.

### 13.13.7 Signature Field

The only way the LAN controllers can tell if a shared SPI Flash is programmed, is by trying to read the shared SPI Flash. The LAN controllers first read the Control word at word address 0x0 and 0x2000. They then check the signature value at bits 7 and 6 in both addresses.

If bit 7 is 0b and bit 6 is 1b in (at least) one of the two addresses, it considers the shared SPI Flash to be present and valid. It then reads the additional shared SPI Flash words from that section and programs its internal registers based on the values read. Otherwise, it ignores the values read from that location and does not read additional words.

If the signature bits are valid at both addresses, the LAN controllers assume that the first section is the valid one.

### 13.13.8 VPD Support

The shared SPI Flash image can contain an area for VPD. This area is managed by the OEM vendor and does not influence the behavior of hardware. Word 0x2F of the shared SPI Flash image contains a pointer to the VPD area in the shared SPI Flash. A value of 0xFFFF means VPD is not supported and the *PCI_CAPCTRL.VPD_EN* bit should be cleared in the PCI shared SPI Flash section (described in the PRM), to prevent the VPD capability from appearing in the configuration space.

The maximum area size is 1024 bytes but can be smaller. The VPD block is built from a list of resources. A resource can be either large or small. The structure of these resources are listed in Table 13-22, "Small Resource Structure" and Table 13-23, "Large Resource Structure".

**Table 13-22. Small Resource Structure**

| Offset | 0 | 1 — n |
|---|---|---|
| Content | Tag = 0xxx,xyyyb (Type = Small(0), Item Name = xxxx, length = yy bytes) | Data |

**Table 13-23. Large Resource Structure**

| Offset | 0 | 1 — 2 | 3 — n |
|---|---|---|---|
| Content | Tag = 1xxx,xxxxb (Type = Large(1), Item Name = xxxxxxxx) | Length | Data |

The LAN controllers parse the VPD structure during the auto-load process following PCIe reset in order to detect the read only and read/write area boundaries. The LAN controllers assume the following VPD fields with the limitations listed.

**Table 13-24. VPD Structure**

| Tag | Length (bytes) | Data | Resource Description |
|---|---|---|---|
| 0x82 | Length of identifier string | Identifier | Identifier string. |
| 0x90 | Length of RO area | RO data | VPD-R list containing one or more VPD keywords. |
| 0x91 | Length of RW area | RW data | VPD-W list containing one or more VPD keywords. This part is optional. |
| 0x78 | N/A | N/A | End tag. |

VPD structure limitations:

- The structure must start with a Tag = 0x82. If the LAN controllers do not detect a value of 0x82 in the first byte of the VPD area or the structure does not follow the description of the VPD structure, they assume the area is not programmed and the entire 1024 bytes area is read only.
- The RO area and RW area are both optional and can appear in any order. A single area is supported per tag type. Refer to Appendix I in the PCI 3.0 specification for details of the different tags.
- If a VPD-W tag is found, the area defined by its size is writable via the VPD structure.
- The structure must end with a Tag = 0x78. The tag must be word aligned.
- The VPD area can be accessed through the PCIe configuration space VPD capability structure listed in the VPD structure. Write accesses to a read-only area or any access to an offset outside of the VPD area via this structure are ignored.
- VPD area must be mapped in the first 16 KB section of the shared SPI Flash mapped to the shadow RAM.
- VPD software does not check the semaphores before attempting to access the shared SPI Flash via dedicated VPD registers. Even if the shared SPI Flash is owned by another entity, VPD software read access to the VPD area in the shared SPI Flash might complete immediately since it is first performed against the shadow RAM. However, VPD software write access might not complete immediately since the VPD changes are committed to the shared SPI Flash device at the LAN controller initiative, once the other entity releases shared SPI Flash ownership, which can take up to several seconds.

### 13.13.8.1 VPD Access Flows

#### 13.13.8.1.1 First VPD Area Programming

The VPD capability is exposed in the PCIe configuration space only if the *PCI_CAPCTRL.VPD_EN* bit is set, regardless to any other sanity check that is performed on the VPD area contents.

The VPD content and pointer can be written on a blank shared SPI Flash without any limitation, such as for any other shared SPI Flash module when in the blank shared SPI Flash programming mode. After protection is enabled, if *VPD Write Enable* bit in shared SPI Flash control word 1 is cleared, only the RW area of the VPD is writable and only via the VPD interface.

### 13.13.8.2 VPD Area Update Flows

1. The host initiates a VPD write by programming the offset and data fields of the VPD capability register set and then setting the capability's *Flag* bit. (bit 15 in VPD Address Register - 0xE2).
2. Firmware checks that the VPD write is allowed - it checks that the write offset falls within the VPD-RW area. If writing is not allowed, firmware clears the VPD flag in the configuration space to notify the host VPD software that the transaction completed. It then exits the flow.
3. Firmware indicates VPD access completion by clearing the VPD flag in the configuration space.

### 13.13.8.3 Shared SPI Flash Update Read, Write, and Erase Sequences

Refer to for the flow required to update secure shared SPI Flash modules.

Any software flow described in this section must be preceded by taking shared SPI Flash ownership via semaphores.

### 13.13.8.4 Software Flow to the Bit Banging Interface

To directly access the Flash when Flash is blank or not protected, software should follow these steps:

1. Write a 1b to the *Flash Request* bit (*FLA.FL_REQ*).
2. Poll the *Flash Grant* bit (*FLA.FL_GNT*) until it becomes 1b. It remains 0b as long as there are other accesses to the shared SPI Flash.
3. Write or read the shared SPI Flash using the direct access to the 4-wire interface as defined in the FLA register. The exact protocol used depends on the shared SPI Flash placed on the board.
4. Write a 0b to the *Flash Request* bit (*FLA.FL_REQ*).
5. Following a write or erase instruction, software should clear the *Request* bit only after it has checked that the cycles were completed by the shared SPI Flash. This can be checked by reading the *BUSY* bit in the shared SPI Flash device STATUS register. Refer to shared SPI Flash data sheet for the opcode to be used for reading the Status register.

*Note:* The bit banging interface is blocked during normal operation (protection enabled). Software should use the *FLSW* mode. Firmware can use this interface at all times.

#### 13.13.8.5 Erase Flow Using the FLA Register

To directly erase a sector in the shared SPI Flash when the shared SPI Flash is blank or not protected, software should follow these steps:

1. Take ownership of shared SPI Flash update semaphore.
2. Set the *Flash Address* (*FLA.FL_ADDR*) to the index of the 4 KB sector to erase and *Sector Erase* bit (*FLA.FL_SER*) bit.
3. Read the FLA register until *Flash Busy* bit (*FLA.FL_BUSY*) is cleared.
4. Release ownership of shared SPI Flash update semaphore.

To directly erase the entire shared SPI Flash update when shared SPI Flash update is blank or not protected, software should follow these steps:

1. Take ownership of shared SPI Flash update semaphore.
2. Set *Device Erase* bit (*FLA.FL_DER*) bit.
3. Read the FLA register until *Flash Busy* bit (*FLA.FL_BUSY*) is cleared.
4. Release ownership of shared SPI Flash semaphore.

#### 13.13.8.6 Software Access flow to Shadow RAM

##### 13.13.8.6.1 Read Interface

Software can read from the shadow RAM using the following flow:

1. Send a Shadow RAM Read Host Interface command (described in the PRM) with the address and length to read.
2. Wait for the command to complete
3. Read the data from the response buffer.

##### 13.13.8.6.2 Write Interface

Software can write to the shadow RAM using the following flow:

1. Send a Shadow RAM Write host Interface command (described in the PRM) with the address, length and data to write.
2. Wait for the command to complete

### 13.13.8.7 Flash Access Flow via the Memory Mapped Interface

#### 13.13.8.7.1 Read Access

Software can always use the Flash BAR for read accesses.

*Note:* Software should take semaphore ownership before executing the flow.

#### 13.13.8.7.2 Write Access

When the shared SPI Flash is blank or protection is disabled, software might initiate a write cycles via the Flash BAR as follows:

1. Take semaphore ownership before executing the flow.
2. Write the data byte to the shared SPI Flash through the Flash BAR.
3. Poll the *FL_BUSY* flag in the FLA register until cleared.
4. Repeat steps 2 and 3 to write additional bytes.
5. Release shared SPI Flash semaphore ownership.

As a response, hardware executes the following steps for each write access:

1. Set the *FL_BUSY* bit in the *FLA* register.
2. Initiate autonomous write enable instruction.
3. Initiate the program instruction right after the enable instruction.
4. Poll the Flash status until programming completes.
5. Clear the *FL_BUSY* bit in the *FLA* register.

*Note:* Software must erase the sector prior to programming it.

### 13.13.8.8 Software Flash/Shadow RAM Program Flow

Software must take semaphore ownership before executing the flow.

Software can write to non-write protected areas of the shared SPI Flash using the following flow:

1. Send a Flash Write Host Interface command (described in the PRM) with the address, length and data to write.
2. Wait for the command to complete.

### 13.13.8.9 Software Flash Read Flow via the Flash-mode Interface

Software must take semaphore ownership before executing the flow.

Software can read from the Flash using the following flow:

1. Send a Flash Read host Interface command (described in the PRM) with the address and length of the data to read.
2. Wait for the command to complete.
3. Read the data from the completion of the command.

## 13.13.9 Extended Shared SPI Flash Update Flows

### 13.13.9.1 Flow for Updating Secured Modules

This section describes the flow to use to update the firmware image.

In order to protect the shared SPI Flash update procedure from power-down events, a double image policy is used for each of the updated modules. The software flow to update a module is as follows:

1. Take ownership over the shared SPI Flash update via the semaphore bits. See PRM.
   a. If *SW_FW_SYNC.NVM_Update_Started* bit is read as clear, then set this bit together with setting the shared SPI Flash semaphore bit. It is used to notify other entities that a long shared SPI Flash update process that might take up to several minutes has started. During this time, other entities cannot perform a write access to the firmware module, but reading this module in between update write bursts is allowed using the shared SPI Flash memory mapping.
   b. Otherwise, release the shared SPI Flash semaphore ownership and restart the update process later on.
2. Read the pointer to the free provisioning area (shared SPI Flash word 0x40). Check that the free provisioning area size read from shared SPI Flash word 0x41 is greater or equal to the size of the new firmware image to be loaded in the shared SPI Flash.
   a. If not, release the shared SPI Flash semaphore ownership, clear the *SW_FW_SYNC.NVM_Update_Started* bit and exit the flow.
3. Initiate sector erase instructions (Section 13.13.8.3) to the entire free space provisioning segment.
   a. In order to guaranty shared SPI Flash semaphore ownership time does not exceed the one second timeout, it is recommended to perform at this step no more than four 4 KB sector erase operations at once in a burst, releasing semaphore ownership for 10 ms in between. This way, other entities can insert shared SPI Flash read accesses in between bursts without waiting for the entire update process completion, which might take minutes.
4. Write the new firmware/option ROM to the free provisioning area via Flash-mode access (Section 13.13.8.9)
   a. Same as step 3a, it is recommended to write at this step no more than four 4 KB sectors at once in a burst, releasing semaphore ownership for 10 ms in between.
5. Send a shared SPI Flash module update module ID of the section to update.The encoding of the modules is:

### Table 13-25. SPI Flash Module Encoding

| Module | ID | Reference |
|---|---|---|
| Firmware code | 0x1 | |
| Reserved | 0x2 | |
| Reserved | 0x3 | |
| Reserved | 0x4 | |
| Reserved | 0x5 | |
| Reserved | 0x6:0xFD | |
| Reserved | 0xFE | |
| Reserved (full shadow RAM) | 0xFF | |

6. Release the shared SPI Flash semaphore and clear the *SW_FW_SYNC.NVM_Update_Started* bit.
   a. Software must avoid taking the shared SPI Flash semaphore again until the firmware command has completed. Any attempt to write the shared SPI Flash until then is not performed by the device.
7. Firmware swaps between the free provisioning area pointer (word 0x40) and the firmware code module pointer, Network boot module pointer or the PHY pointer located at the shadow RAM word address 0x3A/0x5/0x4, respectively. Firmware dumps the shadow RAM into the shared SPI Flash.
8. Software waits for the command to complete.
   a. If the update process failed due to a security check failure or a shared SPI Flash write fault, an Authentication Error (0x80) or Data Error (0x6), respectively is returned. Software must then exit the flow, prior to attempting another update.

## 13.14 Configurable I/O Pins — Software-Definable Pins (SDPs)

The LAN controller has two software definable pins (SDP pins) per port that can be used for miscellaneous hardware or software-controllable purposes. Unless specified otherwise, these pins and their function are bound to a specific LAN device. The use, direction, and values of SDP pins are controlled and accessed by the Extended SDP Control (ESDP) register. To avoid signal contention, following power up, all SDP pins are defined as input pins.

*Note:* LAN0 has 4 SDP per port (total 8) and LAN1 has 2 SDP per port (total 4).

For the LAN controllers, the primary uses for the SDPs are:

- MOD_ABS connection pin for use in SFI supported designs where optics are used (SDP0 only)
- IEEE 1588 support
- Custom driver designs that require the use of an SDP for a desired use
- Creation of extended bus like I2C for LAN1 (created by MUXing pins)

Some SDP pins have specific functionality:

- The default direction of the SDP pins is loaded from the SDP Control word in the shared SPI Flash.
- The SDP pins can also be configured for use as External Interrupt Sources (GPI). To act as GPI pins, the desired pins must be configured as inputs and enabled by the GPIE register. The LAN controller can apply an inversion to the Rx data path via a programmable register. When detected, a corresponding GPI interrupt is indicated in the EICR register.

*Note:* An SDP configured as output can also generate interrupts, but this is not a recommended configuration.The bit mappings are listed in the following table for clarity.

### Table 13-26. GPI to SDP Bit Mappings

| SDP Pin To Be Used As GPI | ESDP Field Settings | | Resulting EICR Bit (GPI) |
|---|---|---|---|
| | Directionality | Enable as GPI interrupt | |
| 3 | SDP3_IODIR | SDP3_GPIEN | 28 |
| 2 | SDP2_IODIR | SDP2_GPIEN | 27 |
| 1 | SDP1_IODIR | SDP1_GPIEN | 26 |
| 0 | SDIP_IODIR | SDP0_GPIEN | 25 |

- The lowest SDP pins (SDP0_0) of port 0 can be used to encode the NC-SI package ID of the LAN controller. This ability is enabled by setting bit 15 in NC-SI Configuration 2 word (offset 0x07) of the shared SPI Flash. The 3-bit package ID is encoded as follows: Package ID = {0, SDP0_0, 0}.

When the SDP pins are used as IEEE1588 auxiliary signals they can generate an interrupt on any transition (rising or falling edge).

All SDP pins can be allocated to hardware functions. See more details on IEEE1588 auxiliary functionality in the PRM while I/O pin functionalities are programmed by the TimeSync Auxiliary Control (TSAUXC) register.

If mapping of these SDP pins to a specific hardware function is not required then the pins can be used as general purpose software defined I/Os. For any of the function-specific usages, the SDP I/O pins should be set to native mode by software setting of the SDPxxx_NATIVE bits in the ESDP register. Native mode in those SDP I/O pins, defines the pin functionality at inactive state (reset or power down) while behavior at active state is controlled by the software. The hardware functionality of these SDP I/O pins differ mainly by the active behavior controlled by software.

The following table lists the setup required to achieve each of the possible SDP configurations.

### Table 13-27. SDP Settings

| SDP | Usage | Shared SPI Flash Setting | ESDP | | |
|---|---|---|---|---|---|
| | | | SDPx_NATIVE | SDPx_IODIR | SDP1_Function |
| 0 | SDP | N/A | 0b | Input/Output | N/A |
| | MOD_ABS for SFI | bit 15 in NC-SI Configuration 2 shared SPI Flash word | 0b | Input | |
| | 1588 functionality as defined by the TSSDP register | N/A | 1b | Input/Output | |
| 1 | SDP. | N/A | 0b | Input/Output | |
| | XXXXXX | bit 2 in PCIe Control 3 word | N/A | Input | |
| | 1588 functionality as defined by the TSSDP register | N/A | 1b | Input/Output | 0b |

## 13.15 LEDs

The LAN controller implementation has two devices (LAN0 and LAN1) that provide two ports per device. LAN0 Port 0 has four and the other three ports have two output drivers intended for driving external LED circuits. Each of the LED outputs can be individually configured to select the particular event, state, or activity, indication on that output. In addition, each LED can be individually configured for output polarity as well as for blinking versus non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the LEDCTL register. Furthermore, the hardware-default configuration for all LED outputs can be specified via shared SPI Flash fields thereby supporting LED displays configurable to a particular OEM preference.

Each of the LEDs can be configured to use one of a variety of sources for output indication. For more information on the MODE bits refer to the LEDCTL register description in the PRM.

The *IVRT* bits enable the LED source to be inverted before being output or observed by the blink-control logic. LED outputs are assumed to normally be connected to the negative side (cathode) of an external LED.

The *BLINK* bits control whether the LED should be blinked (on for 200 ms, then off for 200 ms) while the LED source is asserted. The blink control can be especially useful for ensuring that certain events, such as ACTIVITY indication, cause LED transitions, which are sufficiently visible by a human eye.

*Note:* The LINK/ACTIVITY mode ignores the BLINK value. The LED behavior in this mode is:

- Off if there is no LINK
- On if there is LINK and no ACTIVITY
- Blinks if there is LINK and ACTIVITY

The following mapping is used to specify the LED control source (MODE) for each LED output:

### Table 13-28. LED Control Source (MODE) Mapping

| Mode | Selected Mode | Source Indication |
|---|---|---|
| 0000b | LINK_UP | Asserted or blinking according to the LEDx_BLINK setting when any speed link is established and maintained. |
| 0001b | LINK_10G | Asserted or blinking according to the LEDx_BLINK setting when a 10 Gb/s link is established and maintained. |
| 0010b | MAC_ACTIVITY | Active when link is established and packets are being transmitted or received. In this mode, the LEDx_BLINK must be set. |
| 0011b | FILTER_ACTIVITY | Active when link is established and packets are being transmitted or received that passed MAC filtering. In this mode, the LEDx_BLINK must be set. |
| 0100b | LINK/ACTIVITY | Asserted steady when link is established and there is no transmit or receive activity. Blinking when there is link and receive or transmit activity. |
| 0101b | LINK_1G[1] | Asserted or blinking according to the LEDx_BLINK setting when a 1 Gb/s link is established and maintained. |
| 0110b | Reserved[2] | Reserved. |
| 0111b | LINK_2_5G | Asserted or blinking according to the LEDx_BLINK setting when a 2.5 Gb/s link is established and maintained. |
| 1000b | Reserved | Reserved |
| 1110b | LED_ON | Always asserted or blinking according to the LEDx_BLINK setting. |
| 1111b | LED_OFF | Always deasserted. |

***Notes:***
1. 10/100Mb operation will take same LED operation as 1G where applicable.
2. Undefined values are reserved.

## 13.16 Network Management Interface (MDIO or $I^2C$)

The LAN controllers support MDIO (Section 13.5.2, "Interconnect Interfaces") and/or $I^2C$ (Section 13.5.2, "Interconnect Interfaces") interfaces, for control plane connection between the MAC (master side) and external PHY devices. The management interface enables both MAC and software access to the PHY for monitoring and controlling of the PHYs functionality and configurations. The LAN controllers are compliant with the IEEE Std 802.3 Clause 45 in LAN and 1 GbE operation. The LAN controllers also support IEEE Std 802.3 Clause 22 frame formats and register address space for accessing legacy PHY registers.

There are various connectivity options to external PHYs some use MDIO, such as Base-T PHYs, others might use $I^2C$, like SFP+ modules while in some special scenarios both might be needed such as when a device does not natively support SFI and requires the system designer to use both an external PHY for SFI translation, and an external SFP+ module. In such scenarios, the LAN controllers might need to use the MDIO to manage the external PHY and the $I^2C$ to manage the external SFP+ module.

Make sure you select external devices that have been tested by Intel and that are supported in our native driver and tools. If an external device is selected that is not supported natively, it will be the customer's responsibility to change the driver, maintain updates to end users, and perform validation functions related to this added interface.

The list of supported devices can be found in Table 13-3, "Supported System Configurations".

### 13.16.1 $I^2C$ or MDIO Selection

Since the LAN controller supports both MDC/MDIO and $I^2C$ management interfaces, the final board and device setup must be communicated to the software device driver via a shared SPI Flash loaded configuration.

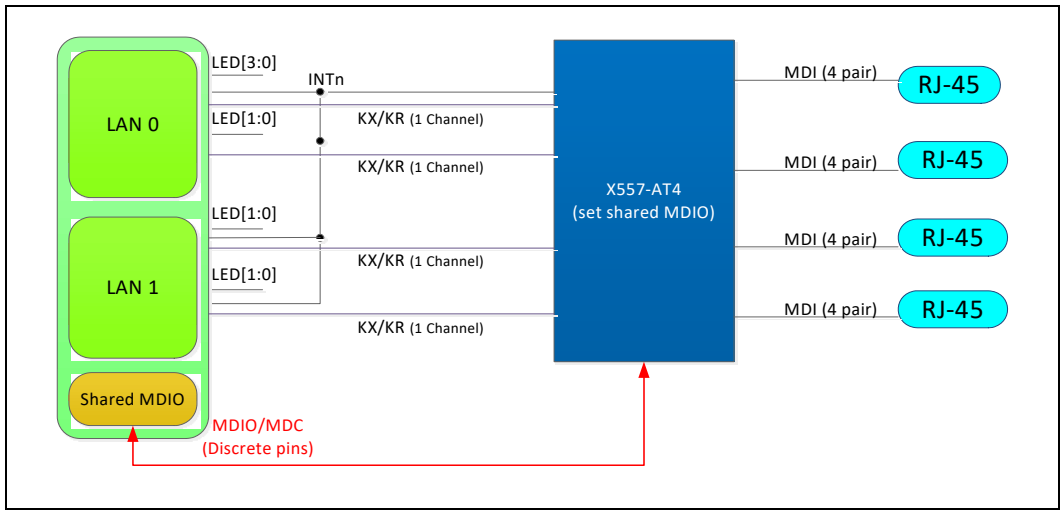
The following settings are provided in the NW Management Interface Select register for the software device driver to read. Based on these settings, the software device driver is able to understand if there is a companion device connected through either of the port's management interfaces. After power on, the software device driver reads the identity of the device connected through the interface and learns the physical topology.

Figure 13-5, "10GBASE-T Implementation" through Figure 13-8, "1G BASE-T Implementation" show the possible connectivity options.

## 13.16.2 Recommended Networking Topologies

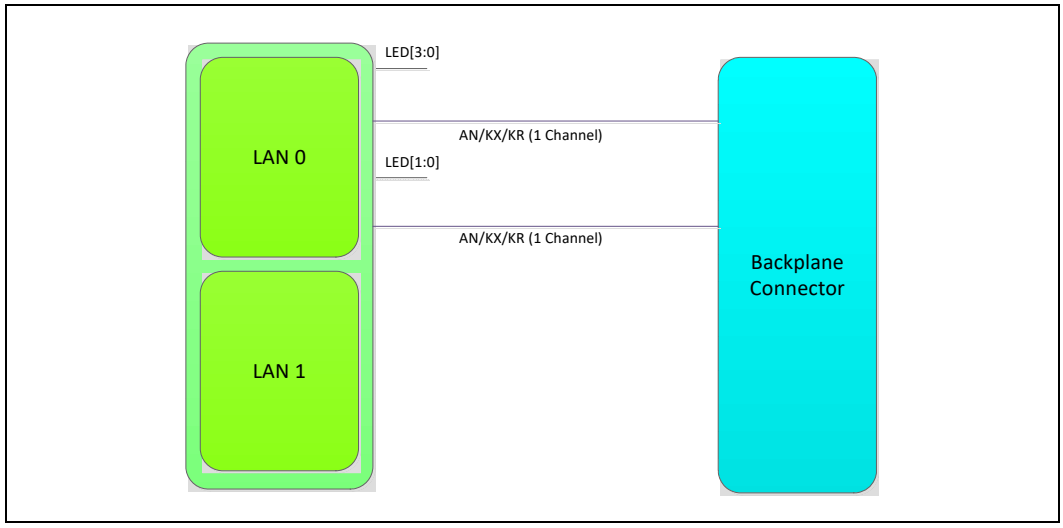
The following diagrams show various recommended options for LAN controllers support. Each of these implementations are described in detail in Table 13-3, "Supported System Configurations".

**Figure 13-5. 10GBASE-T Implementation**




This implementation uses a 10GBASE-T PHY to enable four ports of 10GBASE-T via the KR support found in the LAN controller LAN0 and LAN1. MDIO/MDC is used to configure the PHY at initialization for KR mode. Refer to the list of supported external PHYs in Table 13-3, "Supported System Configurations".

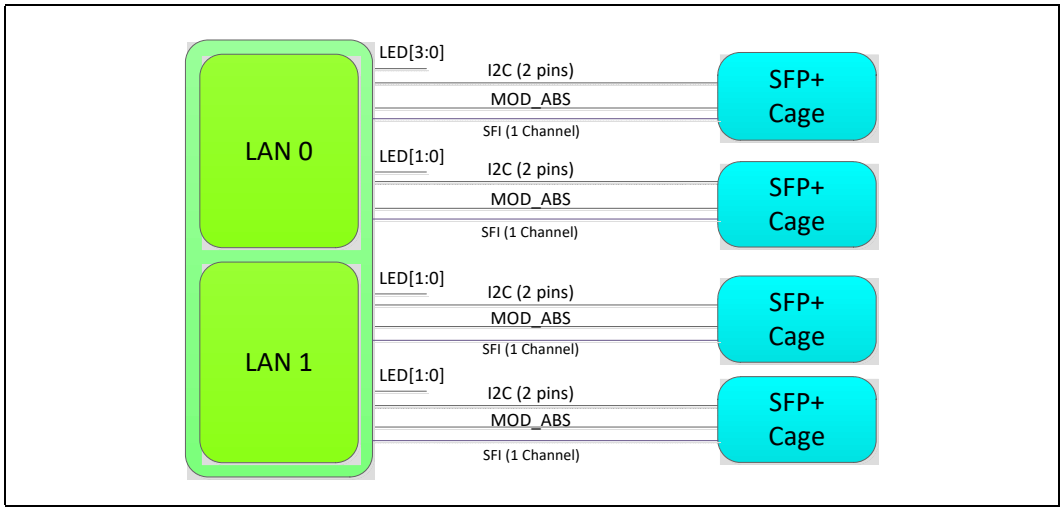
**Figure 13-6. 10G Backplane Implementation Using KR to Backplane (Switch)**




*Note:* Shown is LAN0 Ports 0 and 1 connected to the backplane. Similar connections would be made if LAN1 connectivity was desired.

This implementation uses the KR support found in the LAN controller LAN0 and LAN1 (all 4 ports) for connection to a supported backplane partner. Refer to the list of supported switches in Table 13-3, "Supported System Configurations".

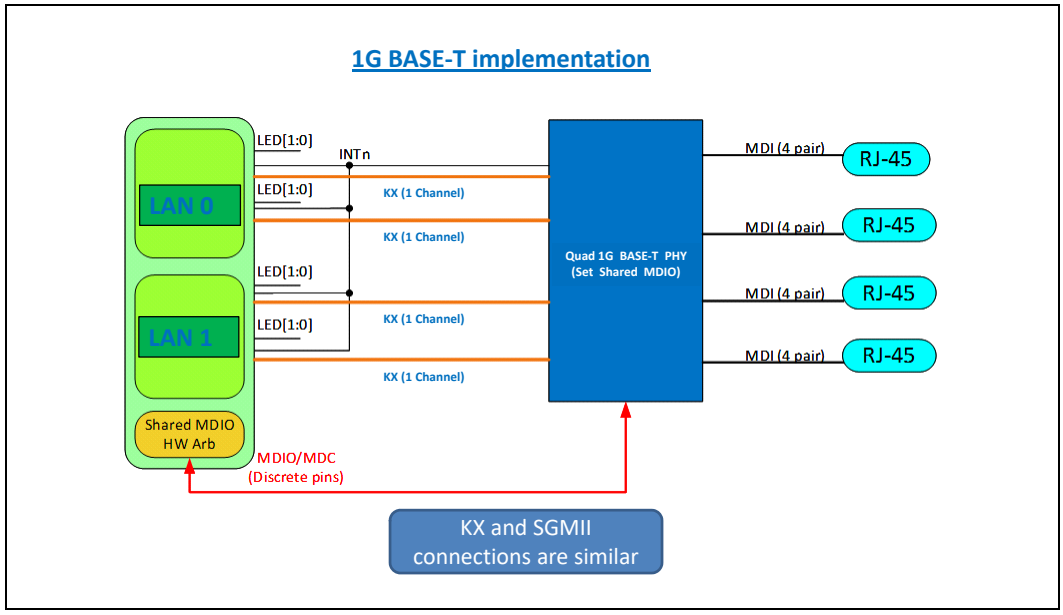
**Figure 13-7. SFP+ Implementation**




This implementation uses native SFI found in the LAN controller LAN0 and LAN1 for connection to a supported SFP+ optical modules. MOD_ABS (module absence) is provided via SDP support. $I^2C$ connectivity is needed to ensure supported optical modules are populated.

## 13.16.3 Management Data I/O Interface (MDIO)

**Figure 13-8. 1G BASE-T Implementation**



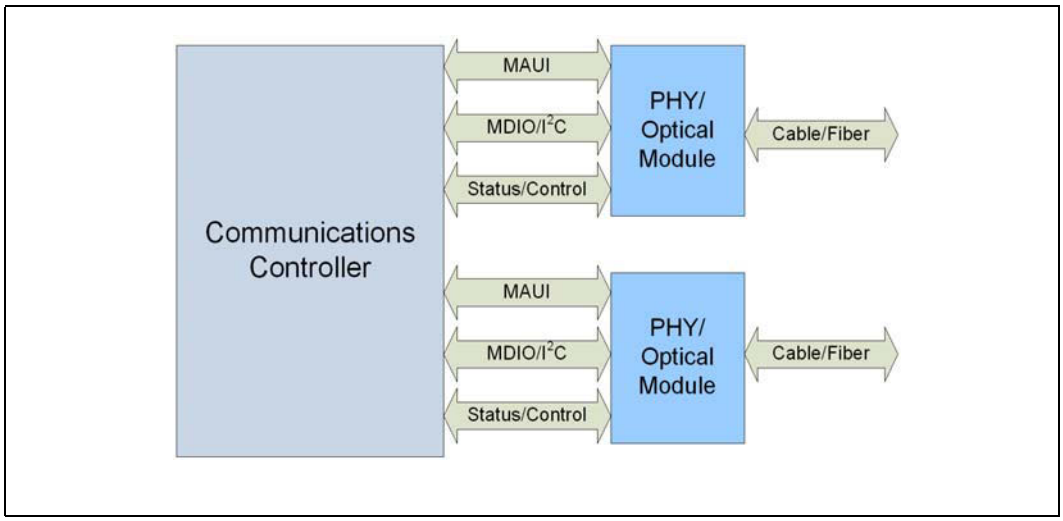

The LAN controllers support the MDIO interface for a control plane connection between the MAC (master side) and PHY devices. The MDIO interface enables both MAC and software access to the PHY for monitoring and controlling of the PHYs functionality and configurations. The LAN controllers are compliant with the IEEE802.3 clause 45 in both 10 GbE and 1 GbE operation. The LAN controllers also support IEEE 802.3 clause 22 frame formats and register address space for accessing legacy PHY registers. Note that these pins are shared with the LED pins and at a given time only one of them is functional based on an shared SPI Flash configuration.

*Note:* The MDIO interface uses LVTTL signaling as defined in Clause 22 of the IEEE802.3 standard. To access PHYs that support clause 45 1.2V electrical interface, level translators might be needed on board. The MDIO/MDC connection is not required for KX connectivity.

Figure 13-9 shows the basic connectivity between the PHY and MAC.

*Note:* The MAUI is defined as Medium Attachment Unit Interface.

**Figure 13-9. Basic PHY/MAC Connectivity**




The MDIO interface is a simple 2-wire serial interface between the MAC and PHY and is used to access Control and Status registers inside the PHY. The interface is implemented using two LVTTL I/O signals:

1. MDC — MDIO-interface clock signal driven by an external MAC (STA) device.
2. MDIO — Read/write data between an external MAC and PHY

### 13.16.3.1 MDIO Timing Relationship to MDC

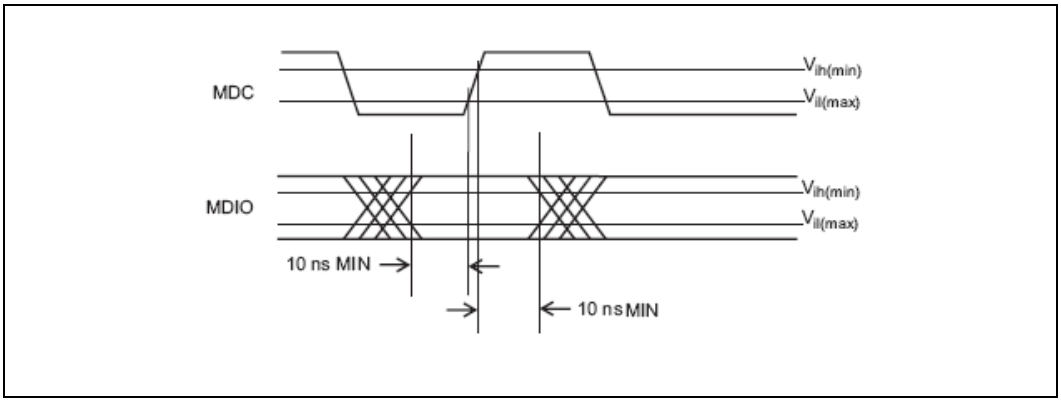
The MDC clock toggles during a read/write operation at a frequency of 2.4 MHz or 240 KHz depending on the link speed and register bit HLREG0.MDCSPD as listed in Table 13-29, "MDC Frequency as Function of Link Speed and MDC Speed Bit".

**Table 13-29. MDC Frequency as Function of Link Speed and MDC Speed Bit**

| Link Speed | MDCSPD=1b | MDCSPD=0b |
|---|---|---|
| 10 Gb/s | Reserved | 2.4 MHz |
| 1 Gb/s | 2.4 MHz | 240 KHz |

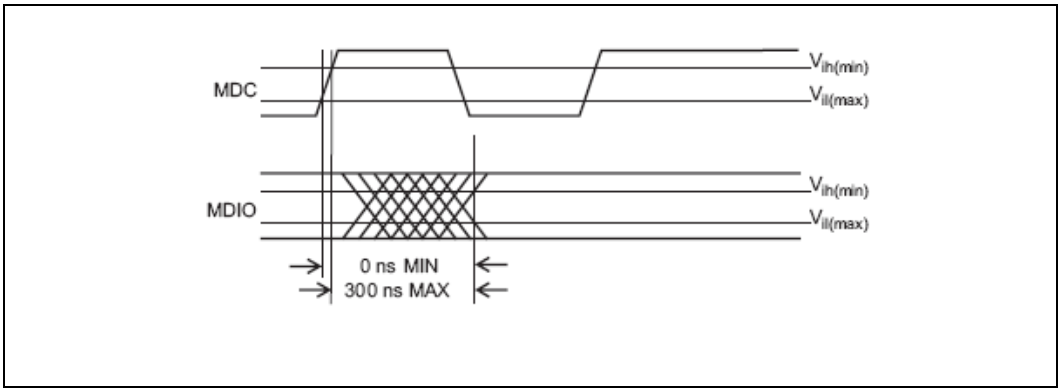
MDIO is a bidirectional signal that can be sourced by the Station Management Entity (STA) or the PHY. When the STA sources the MDIO signal, the STA must provide a minimum of 10 ns of setup time and a minimum of 10 ns of hold time referenced to the rising edge of MDC, as shown in Figure 13-10, "MDIO Timing Source by the MAC" (measured at the MII connector).

**Figure 13-10. MDIO Timing Source by the MAC**




When the MDIO signal is sourced by the PHY, it is sampled by the MAC (STA) synchronously with respect to the rising edge of MDC. The clock to output delay from the PHY, as measured at the MII connector, must be a minimum of 0 ns, and a maximum of 300 ns, as shown in Figure 13-11, "MDIO Timing Sourced by the PHY".

**Figure 13-11. MDIO Timing Sourced by the PHY**



### 13.16.3.2 IEEE802.3 Clause 22 and Clause 45 Differences

IEEE802.3 clause 45 provides the ability to access additional device registers while still retaining logical compatibility with interface defined in Clause 22. Clause 22 specifies the MDIO frame format and uses an ST code of 01 to access registers. In clause 45, additional registers are added to the address space by defining MDIO frames that use a ST code of 00.

Clause 45 (MDIO interface) major concepts:

- Preserve management frame structure defined in IEEE 802.3 Clause 22.
- Define mechanism to address more registers than specified in IEEE802.3 Clause 22.
- Define ST and OP codes to identify and control the extended access functions.

### 13.16.3.3 MDIO Management Frame Structure

The MDIO interface frame structure defined in IEEE802.3 clause 22 and Clause 45 are compatible so that the two systems supporting different formats can co-exist on the same MDIO bus. The SoC LAN controllers support both frame structures to enable interfacing PHYs that support either protocol.

The basic frame format as defined in IEEE802.3 clause 22 can optionally be used for accessing legacy PHY registers is listed in Table 13-30, "Clause 22 Basic MDIO Frame Format".

**Table 13-30. Clause 22 Basic MDIO Frame Format**

| Function | Management Frame Fields | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Frame | Pre | ST | OP | PRTAD | REGAD | TA | Data | Idle |
| Read | 1...1 | 01 | 10 | PPPPP | RRRRR | Z0 | DDDDDDDDDDDDDDDD | Z |
| Write | 1...1 | 01 | 01 | PPPPP | RRRRR | 10 | DDDDDDDDDDDDDDDD | Z |

The MDIO interface defined in clause 45 uses indirect addressing to create an extended address space enabling access to a large number of registers within each MDIO Managed Device (MMD). The MDIO management frame format is listed in Table 13-31, "Clause 45 Indirect Addressing MDIO Frame Format".

### Table 13-31. Clause 45 Indirect Addressing MDIO Frame Format

| Function | Management Frame Fields | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Frame | Pre | ST | OP | PRTAD | DEVAD | TA | Address / Data | Idle |
| Address | 1...1 | 00 | 00 | PPPPP | EEEEE | 10 | AAAAAAAAAAAAAAAA | Z |
| Write | 1...1 | 00 | 01 | PPPPP | EEEEE | 10 | DDDDDDDDDDDDDDDD | Z |
| Read | 1...1 | 00 | 11 | PPPPP | EEEEE | Z0 | DDDDDDDDDDDDDDDD | Z |
| Post-Read Increment Address | 1...1 | 00 | 10 | PPPPP | EEEEE | Z0 | DDDDDDDDDDDDDDDD | Z |

To support clause 45 indirect addressing each MMD (PHY — MDIO managed device) implements a 16-bit address register that stores the address of the register to be accessed by data transaction frames. The address register must be overwritten by address frames. At power up or device reset, the contents of the address register are undefined. Write, read, and post-read-increment-address frames must access the register whose address is stored in the address register. Write and read frames must not modify the contents of the address register. Upon receiving a post-read-increment-address frame and having completed the read operation, the MMD increments the Address register by one (up to a value of 0xFFFF). Each MMD supported implements a separate address register, so that the MMDs address registers operate independently of one another.

Idle Condition (IDLE) — The IDLE condition on MDIO is a high-impedance state. All three state drivers must be disabled and the PHYs pull-up resistor pulls the MDIO line to a logic one.

Preamble (PRE) — At the beginning of each transaction, the station management entity must send a sequence of 32 contiguous consecutive one bits on MDIO with 32 corresponding cycles on MDC to provide the PHY with a pattern that it can use to establish synchronization. A PHY must observe a sequence of 32 contiguous consecutive one bits on MDIO with 32 corresponding cycles on MDC before it responds to any transaction.

Start of Frame (ST) — The ST is indicated by:

- <00> pattern for clause 45 compatible frames for indirect access cycles.
- <01> pattern for clause 22 compatible frames for direct access cycles.

These patterns ensure a transition from the default value of one on the MDIO signal, and identifies the start of frame.

Operation Code (OP) — The *OP* field indicates the type of transaction being performed by the frame.

For Clause 45 compatible frames:

- A <00> pattern indicates that the frame payload contains the address of the register to access.
- A <01> pattern indicates that the frame payload contains data to be written to the register whose address was provided in the previous address frame.
- A <11> pattern indicates that the frame is an indirect read operation.
- A <10> pattern indicates that the frame is an indirect post-read-increment-address operation.

For Clause 22 compatible frames:

- A <10> pattern indicates a direct read transaction from a register.
- A <01> pattern indicates a direct write transaction to a register.

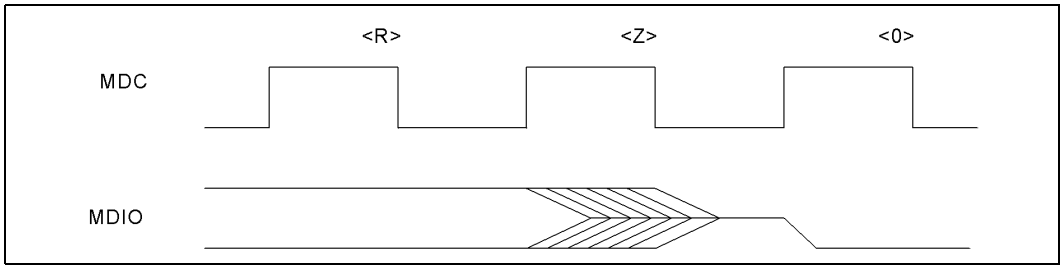
**Port Address (PRTAD)** — The PRTAD is five bits, allowing 32 unique PHY port addresses. The first *PRTAD* bit to be transmitted and received is the MSB of the address. A station management entity must have prior knowledge of the appropriate port address for each port to which it is attached, whether connected to a single port or to multiple ports.

**Device Address (DEVAD)** — The DEVAD is five bits, allowing 32 unique MMDs per port. The first *DEVAD* bit transmitted and received is the MSB of the address. This field is relevant only in clause 45 compatible frames (ST=<00>).

**Register Address (REGAD)** — The REGAD is five bits, allowing 32 individual registers to be addressed within each PHY. The first *REGAD* bit transmitted and received is the MSB of the address. This field is relevant only in clause 22 compatible frames (ST=<01>).

**Turnaround (TA)** — The TA time is a 2-bit time spacing between the *DEVAD* field and the *Data* field of a management frame. This is to avoid contention during a read transaction. For a read or post-read-increment-address transaction, both the STA and the PHY must remain in a high-impedance state for the first bit time of the TA. The PHY must drive a zero bit during the second bit time of the TA of a read or post-read-increment-address transaction. During a write or address transaction, the STA must drive a one bit for the first bit time of the TA and a zero bit for the second bit time of the TA. Figure 13-12, "Behavior of MDIO During TA Field of a Read Transaction" shows the behavior of the MDIO signal during the *TA* field of a read transaction.

**Figure 13-12.Behavior of MDIO During TA Field of a Read Transaction**




- Clause 45 compatible frames have 16-bit address/data fields. For an auto-negotiation address cycle, it contains the address of the register to be accessed on the next cycle. For the data cycle of a write frame, the field contains the data to be written to the register. For a read or post-read-increment-address frame, the field contains the contents of the register. The first bit transmitted and received must be bit 15.
- Clause 22 compatible frames have 16-bit data fields. The first data bit transmitted and received must be bit 15 of the register being addressed.

### 13.16.3.4 MDIO Direct Access

The MDI is accessed through registers MSCA and MSRWD. A single management frame is sent by setting bit MSCA.MDICMD to 1b after programming the appropriate fields in the MSCA and MSRWD registers. The MSCA.MDICMD bit is auto cleared after the read or write transaction completes. To execute clause 22 format write operations, the following steps should be done:

1. Data to be written is programmed in field MSRWD.MDIWRDATA.
2. Register MSCA is initialized with the appropriate control information (start, code, etc.) with bit MSCA.MDICMD set to 1b.
3. Wait for bit MSCA.MDICMD to reset to 0b when indicating that the transaction on the MDIO interface is complete.

The steps for clause 22 format read operations are identical to the write operation except that the data in field MSRWD.MDIWRDATA is ignored and the data read from the external device is stored in register field MSRWD.MDIRDDATA bits. Clause 45 format read/write operations must be performed in two steps. The address portion of the pair of frames is sent by setting register field MSCA.MDIADD to the desired address, field MSCA.STCODE to 00b (start code that identifies clause 45 format), and register field MSCA.OPCODE to 00b (clause 45 address register write operation). A second data frame must be sent after the address frame completes. This second frame executes the write or read operation to the address specified in the PHY address register.

## 13.16.4 I²C

The LAN controllers support 2-wire management interface ($I^2C$) for connectivity to external SFP+ modules. Optical and direct attached copper PHYs use 2-wire management interface ($I^2C$) as described in SFF8431 (SFP+) or SFF8636 (Direct attach Cu).

*Note:* Some external PHYs also support an $I^2C$ management interface instead of MDIO. It is up to the board designer to choose which interface to use. See Section 13.16.1, "$I^2C$ or MDIO Selection".

The $I^2C$ interface operates via the I2CCMD and I2CPARAMS register set. Since this register set can be used by either software or firmware in alternation, its ownership must be acquired/released via the semaphore ownership taking/release flows described in the PRM.

The $I^2C$ interface can be used in two methods, a hardware based access, where the device initiates a transaction following a software device driver request via the I2CCMD register or a software controlled bit banging using the I2CPARAMS register.

### 13.16.4.1 Hardware Based $I^2C$ Access

The following flows should be used to access an $I^2C$ register.

As part of device initialization, or anytime before the actual access, the following parameters should be set:

- *I2CPARAMS.PHYADD* - the address of the device to access.
- *I2CPARAMS.ACCESS_WIDTH* - the width of the data to read or write (byte or word).

To execute a write access, the following steps should be done:

1. Check that register is ready: Poll the *I2CCMD.R* bit until it is read as 1b.
2. Command - The I2CCMD register is initialized with the appropriate PHY register address in the *REGADD* field, the data to write in the *DATA* field and the operation (write) to the *OP* field (0b).
   a. If an interrupt is required, set the *I2CCMD.I* field
3. Check that command is done: Poll *I2CCMD.R* bit until it is read as 1b.
   a. Check that no error is indicated in the *I2CCMD.E* field.

To execute a read access, the following steps should be done:

1. Check that the register is ready: Poll *I2CCMD.R* bit until it is read as 1b.
2. Command - The I2CCMD register is initialized with the appropriate PHY register address in the *REGADD* field, and the operation (read) to the *OP* field (1b).
   a. If an interrupt is required, set the *I2CCMD.I* field
3. Check that command is done: Poll *I2CCMD.R* bit until it is read as 1b.
   a. Check that no error is indicated in the *I2CCMD.E* field.
4. Read the data returned from the *I2CCMD.DATA* field. If a byte access is done (*I2CPARAMS.ACCESS_WIDTH* = 0b), only *DATA[7:0]* is valid.

### 13.16.4.2 Bit Bang Based I²C Access

In this mode, the software device driver controls the I²C interface directly using the I2CPARAMS register according to the following table:

**Table 13-32. Direct I²C Interface Control**

| Pad | Field Controlling The Output Value | Field Reflecting The Input Value | Field Controlling The Output Enable Value[1] |
|---|---|---|---|
| I2C clock | CLK_OUT | CLK_IN | CLK_OE_N |
| I2C data | DATA_OUT | DATA_IN | DATA_OE_N |

1. 0b = Pad is output. 1b = Pad is input.

### 13.16.4.3 Supported Commands

The gray columns that follow denote cycles driven by the I²C device. White columns denote cycles driven by the SoC LAN controllers.

When a word Read command (*I2CPARAMS.ACCESS_WIDTH* = 1b, *I2CCMD.OP* =1b) is given, the following sequence is done by the SoC LAN controllers:

**Table 13-33. I²C Read Transaction - Dummy Write**

| 1 | 7 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|---|
| S | Device Address | Wr | A | Register Address | A |
| | From *I2CCMD.PHYADD* | 0 | 0 | From *I2CCMD.REGADD* | 0 |

**Table 13-34. I²C Read Transaction - Word Read**

| 1 | 7 | 1 | 1 | 8 | 1 | 8 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|
| S | Device Address | Rd | A | Data | A | Data | A | P |
| | From *I2CPARAMS.PHYADD* | 1 | 0 | Stored in *I2CCMD.DATA[7:0]* | 0 | Stored in *I2CCMD.DATA[15:8]* | 0 | |

When a byte read command (*I2CPARAMS.ACCESS_WIDTH* = 0b, *I2CCMD.OP* =1b) is given the following sequence is done by the SoC LAN controllers:

**Table 13-35. I²C Read Transaction - Dummy Write**

| 1 | 7 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|---|
| S | Device Address | Wr | A | Register Address | A |
| | From *I2CPARAMS.PHYADD* | 0 | 0 | From *I2CCMD.REGADD* | 0 |

**Table 13-36. I²C Read Transaction - Byte Read**

| 1 | 7 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|---|
| S | Device Address | Rd | A | Data | A |
| | From *I2CPARAMS.PHYADD* | 1 | 0 | Stored in *I2CCMD.DATA[7:0]* | 0 |

When a word Write command (*I2CPARAMS.ACCESS_WIDTH* = 1b, *I2CCMD.OP* =0b) is given the following sequence is done by the LAN controllers:

### Table 13-37. I²C Write Transaction - Word Write

| 1 | 7 | 1 | 8 | 1 | 1 | 8 | 1 | 8 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| S | Device Address | Wr | Register Address | A | P | Data | A | Data | A | P |
| | From *I2CPARAMS.PHYADD* | 0 | From *I2CCMD.REGADD* | 0 | | From in *I2CCMD.DATA[7:0]* | 0 | From in *I2CCMD.DATA[15:8]* | 0 | |

When a byte write command (*I2CPARAMS.ACCESS_WIDTH* = 0b, *I2CCMD.OP* =0b) is given the following sequence is done by the SoC LAN controllers:

### Table 13-38. I²C Write Transaction - Byte Write

| 1 | 7 | 1 | 8 | 1 | 1 | 8 | 1 |
|---|---|---|---|---|---|---|---|
| S | Device Address | Wr | Register Address | A | P | Data | A |
| | From *I2CPARAMS.PHYADD* | 0 | From *I2CCMD.REGADD* | 0 | | From in *I2CCMD.DATA[7:0]* | 0 |

## 13.17 Network Interface

There are two LAN controllers (LAN0 and LAN1) with two ports each, providing a total of four ports. Each port contains an independent LAN Media Access Control (MAC) that supports a XGMII-like interface link to an internal PHY (see Figure 13-1, "What is Covered in This Chapter").

The internal MAC/PHY combine to support 1GbE (KX and SGMII), 2.5GbE and 10GbE operation through various modes of operation.

Refer to Section 13.2, "Supported Modes of Operation" for all supported modes of Ethernet for use with internal PHYs (native support) and external PHYs.

### 13.17.1 Companion Integrated PHY Register Access

The LAN controllers provide a set of registers, per port, to enable register access to the software device driver and firmware. The access is an indirect access. For example, the software agent writes the address and data into the pair of registers listed in Table 13-39, "PHY Access Registers" and then the LAN controllers translate this into an internal operation to the relevant PHY sub-section.

**Table 13-39. PHY Access Registers**

| Register | Description |
|---|---|
| PHY Indirect Control | This register is used for:<br>• Configuring the address to access.<br>• Report a status at the end of the operation. |
| PHY Indirect Data | • When software writes the data to this register the write operation is triggered.<br>• When software reads from this register the read operation is triggered. |

#### 13.17.1.1 PHY Register Addressing

The PHY registers are addressed using a 16-bit address. Further, the companion integrated PHY is a dual-port block and has its registers divided to three partitions.

**Table 13-40. PHY Registers' Addressing**

| Partition | Address Decoding | Description |
|---|---|---|
| Common Registers | [15:14] - 11<br>[13:0] - register address | Holds global registers with information that is not per port. |
| Port-0 Registers | [15:14] - 01<br>[13:0] - register address | Holds port-0 configuration and status registers. |
| Port-1 Register | [15:14] - 10<br>[13:0] - register address | Holds port-1 configuration and status registers. |

When software or firmware is programming the PHY Indirect Control register with a port register address, it must be aware of the physical port it needs to access and set bits [15:14] of the address accordingly.

In order to identify the port number, software should read the *LAN_ID* field in the LAN controller Device Status register (0x00000008).

### 13.17.1.2 PHY Indirect Read Operation

1. The software agent checks that the PHY *Indirect Control[Busy]*=0b.
2. The software agent writes the following fields:
   a. PHY Select — These are the three MSBs of the address used to select the PHY sub-section.
   b. Addr[15:0] — internal address of register to access.
3. Software performs a read operation from the PHY Indirect Data register (this triggers the read operation).
4. The LAN controllers issue a read message to the PHY.
5. The LAN controllers send a read completion to software. Software might read the PHY *Indirect Control RESP STAT* field as this indicates the status of the operation.

### 13.17.1.3 PHY Indirect Write Operation

1. The software agent checks that the PHY *Indirect Control[Busy]*=0b.
2. The software agent writes the following fields:
   a. PHY Select - these are the three MSBs of the address used to select the PHY sub-section.
   b. Addr[15:0] - Internal address of register to access.
3. The software agent writes the data into the PHY Indirect Data register.
4. The LAN controllers issue a write message.
5. The software agent polls the PHY *Indirect Control[Busy]* field until it is = 0b again.

## 13.17.2 Ethernet Flow Control (FC)

The LAN controllers support flow control as defined in 802.3x, as well as the specific operation of asymmetrical flow control defined by 802.3z.

*Note:* The LAN controllers can either be configured to receive regular FC packets or PFC packets. The LAN controllers do not support receiving both types of packets simultaneously.

FC is implemented to reduce receive buffer overflows, which result in the dropping of received packets. FC also allows for local controlling of network congestion levels. This can be accomplished by sending an indication to a transmitting station of a nearly full receive buffer condition at a receiving station.

The implementation of asymmetric FC allows for one link partner to send FC packets while being allowed to ignore their reception. For example, not required to respond to PAUSE frames.

The following registers are defined for implementing FC using index[0] of each register:

- MAC Flow Control Register (MFLCN) — Enables FC and passing of control packets to the host.
- Flow Control Configuration (FCCFG) — Determines mode for Tx FC (No FC vs. link-based vs. priority-based). Note that if Tx FC is enabled then Tx CRC by hardware should be enabled as well (*HLREG0.TXCRCEN* = 1b).
- Flow Control Source Address Low, High (RAL[0], RAH[0]).
- Flow Control Destination Address Low, High (*FCAMACL, FCAMACH)* — 6-byte FC multicast address.
- Priority Flow Control Type Opcode (PFCTOP) — Contains the type and OpCode values for PFC.
- Flow Control Receive Threshold Low (FCRTL[7:0]) — A set of 13-bit low watermarks indicating receive buffer emptiness. A single watermark is used in link FC mode.
- Flow Control Transmit Timer Value (FCTTV[3:0]) — A set of 16-bit timer values to include in transmitted PAUSE frame. A single timer is used in link FC mode.
- Flow Control Refresh Threshold Value (FCRTV) — 16-bit PAUSE refresh threshold value (in legacy FC FCRTV[0] must be smaller than FCTTV[0]).

### 13.17.2.1 MAC Control Frames and Reception of Flow Control Packets

#### 13.17.2.1.1 MAC Control Frame — Other than FC

The IEEE specification reserved the Ethertype value of 0x8808 for MAC control frames, which are listed in Table 13-41, "MAC Control Frame Format".

**Table 13-41. MAC Control Frame Format**

| | |
|---|---|
| DA | The *Destination Address* field can be an individual or multicast (including broadcast) address. Permitted values for the *Destination Address* field can be specified separately for a specific control OpCode such as FC packets. |
| SA | Port Ethernet MAC Address (six bytes). |
| Type | 0x8808 (two bytes). |
| Opcode | The MAC control OpCode indicates the MAC control function. |
| Parameters | The MAC control *Parameters* field must contain MAC control OpCode-specific parameters. This field can contain none, one, or more parameters up to a maximum of minFrameSize = 20 bytes. |
| Reserved field = 0x00 | The *Reserved* field is used when the MAC control parameters do not fill the fixed length MAC control frame. |
| CRC | Four bytes. |

#### 13.17.2.1.2 Structure of 802.3X FC Packets

802.3X FC packets are defined by the following three fields (see Table 13-42, "802.3X Packet Format"):

1. A match on the six-byte multicast address for MAC control frames or a match to the station address of the device (Receive Address register 0). The 802.3x standard defines the MAC control frame multicast address as 01-80-C2-00-00-01.
2. A match on the *Type* field. The *Type* field in the FC packet is compared against an IEEE reserved value of 0x8808.
3. A match of the MAC control Opcode field has a value of 0x0001.

Frame-based FC differentiates XOFF from XON based on the value of the PAUSE Timer field. Non-zero values constitute XOFF frames while a value of zero constitutes an XON frame. Values in the *Timer* field are in units of pause quanta (such as slot time). A pause quanta lasts 64 byte times, which is converted into an absolute time duration according to the line speed.

*Note:* XON frame signals the cancellation of the pause from that was initiated by an XOFF frame. Pause for zero pause quanta.

**Table 13-42. 802.3X Packet Format**

| | |
|---|---|
| DA | 01_80_C2_00_00_01 (6 bytes). |
| SA | Port Ethernet MAC Address (6 bytes). |
| Type | 0x8808 (two bytes). |
| Opcode | 0x0001 (two bytes). |
| Time | XXXX (two bytes). |
| Pad | 42 bytes. |
| CRC | Four bytes. |

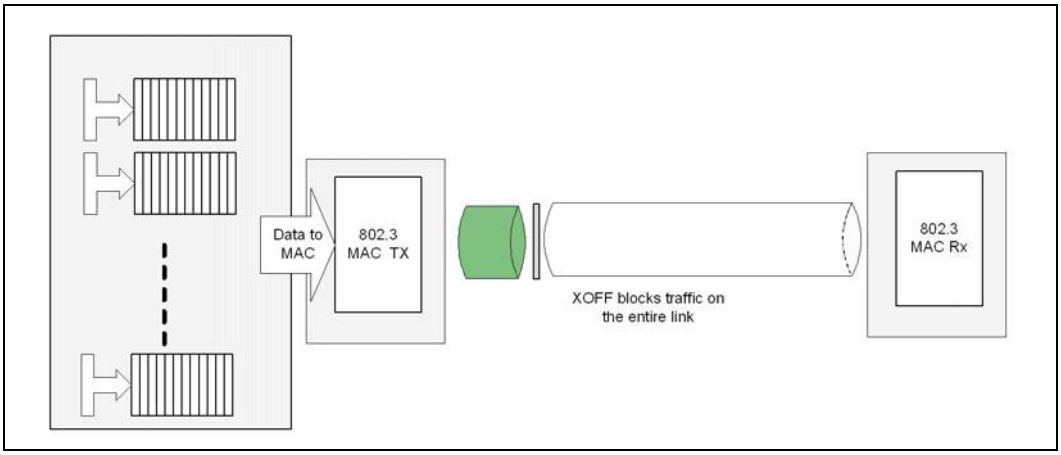
#### 13.17.2.1.3 Priority Flow Control

Support is provided for multiple TCs assigning different priorities and bandwidth per TC. Link-level Flow Control (PAUSE) stops all the TCs. Priority Flow Control (PFC), known as Class Based Flow Control or CBFC, allows more granular Flow Control on the Ethernet link in a DCB environment as opposed to the PAUSE mechanism defined in 802.3X.

PFC is implemented to prevent the possibility of receive packet buffers overflow. Receive packet buffers overflow results in the dropping of received packets for a specific TC. Implement PFC by sending a timer indication to the transmitting station TC (XOFF) of a nearly full receive buffer condition at the LAN controllers. At this point the transmitter stops transmitting packets for that TC until the XOFF timer expires or a XON message is received for the stopped TC.

Similarly, once the LAN controllers receive a priority-based XOFF it stops transmitting packets for that specific TC until the XOFF timer expires or XON packet for that TC is received.

**Figure 13-13.802.3X Link Flow Control (PAUSE)**



#### 13.17.2.1.4 Operation and Rules

The LAN controllers operate in either link FC or in PFC mode. Note that enabling both modes concurrently is not allowed:

- Link FC is enabled by the *RFCE* bit in the MFLCN register.
- PFC is enabled per UP by the corresponding *RPFCE* bit in the MFLCN register, and globally by *MFLCN.RPFCM* bit.

*Note:* Link FC capability must be negotiated between link partners via the auto-negotiation process. The PFC capability is negotiated via some higher level protocol and the resolution is usually provided to the software device driver by the DCB management agent. It is the software device driver responsibility to reconfigure the link FC settings (including RFCE and PRFCE) after the auto-negotiation process was resolved.

*Note:* Receiving a link FC frame while in PFC mode might be ignored or might pause TCs in an unpredictable manner. Receiving a PFC frame while in link FC mode is ignored. Flow control events that are ignored do not increment any flow control statistics counters.

Once the receiver has validated the reception of an XOFF, or PAUSE frame, the device performs the following:

- Increments the appropriate statistics register(s).
- Initialize the pause timer based on the packet's PAUSE *Timer* field (overwriting any current timer value).
  - For PFC, this is done per TC. If several UPs are associated with a TC, then the device sets the timer to the maximum value among all enabled *Timer* fields associated with the TC.
- Disable packet transmission or schedule the disabling of transmission after the current packet completes.
  - For PFC, this is done per paused TC.
  - Tx manageability traffic is bound to a specific TC as defined in the MNGTXMAP register, and is thus paused when its TC is paused.

Resumption of transmission can occur under the following conditions:

- Expiration of the PAUSE timer.
  - For PFC, this is done per TC.
- Receiving an XON frame (a frame with its PAUSE timer set to 0b).
  - For PFC, this is done per TC.

Both conditions clear the relevant TXOFF status bits in the Transmit Flow Control Status (TFCS) register and transmission can resume. Hardware records the number of received XON frames.

#### 13.17.2.1.5 Timing Considerations

When operating at LAN line speed, the LAN controllers must not begin to transmit a (new) frame more than 74 pause quanta after receiving a valid Link XOFF frame, as measured at the wires (a pause quantum is 512 bit times).

When operating at 1 GbE line speed, the LAN controllers must not begin to transmit a (new) frame more than two pause quanta after receiving a valid Link XOFF frame, as measured at the wires.

The 802.1Qbb draft 2.3, proposes that the tolerated response time for priority XOFF frames are the same as Link XOFF frames with extra budget of 19360 bit times if MACSec is used, or of two pause quanta otherwise. This extra budget is aimed to compensate the fact that decision to stop new transmissions from a specific TC must be taken earlier in the transmit data path than for the link flow control case.

### 13.17.2.2 PAUSE and MAC Control Frames Forwarding

Two bits in the Receive Control register control transfer of PAUSE and MAC control frames to the host. These bits are *Discard PAUSE Frames* (*DPF*) and *Pass MAC Control Frames* (*PMCF*). Note also that any packet must pass the L2 filters as well.

- The *DPF* bit controls the transfer of PAUSE packets to the host. The same policy applies to both link FC and PFC packets as listed in Table 13-43, "Transfer of PAUSE Packet to Host (DPF Bit)". Note that any packet must pass the L2 filters as well.
- The Pass MAC Control Frames (PMCF) bit controls the transfer of non-PAUSE packets to the host. Note that when link FC frames are not enabled (RFCE = 0b) then link FC frames are considered as MAC control frames for this case. Similarly, when PFC frames are not enabled (RPFCM = 0b) then PFC frames are considered as MAC control frames as well.

*Note:* When virtualization is enabled, forwarded control packets are queued according to the regular switching procedure.

**Table 13-43. Transfer of PAUSE Packet to Host (DPF Bit)**

| RFCE | RPFCM | DPF | Link FC Handling | PFC Handling |
|---|---|---|---|---|
| 0b | 0b | X | Treat as MAC control (according to *PMCF* setting). | Treat as MAC control (according to *PMCF* setting). |
| 1b | 0b | 0b | Accept. | Treat as MAC control (according to *PMCF* setting). |
| 1b | 0b | 1b | Reject. | Treat as MAC control (according to *PMCF* setting). |
| 0b | 1b | 0b | Treat as MAC control (according to *PMCF* setting). | Accept. |
| 0b | 1b | 1b | Treat as MAC control (according to *PMCF* setting). | Reject. |
| 1b | 1b | X | Unsupported setting. | Unsupported setting. |

### 13.17.2.3 Transmitting PAUSE Frames

The LAN controllers generate PAUSE packets to ensure there is enough space in its receive packet buffers to avoid packet drop. The LAN controllers monitor the fullness of its receive FIFOs and compares it with the contents of a programmable threshold. When the threshold is reached, the LAN controllers send a PAUSE frame. The LAN controllers support both link FC and PFC — but not both concurrently. When DCB is enabled, it only sends PFC, and when DCB is disabled, it only sends link FC.

*Note:* Similar to receiving flow control packets previously mentioned, software can enable FC transmission by setting the *FCCFG.TFCE* field only after it is negotiated between the link partners (possibly by auto-negotiation).

#### 13.17.2.3.1 Priority Flow Control (PFC)

The LAN controllers operate in either a link 802.3X compliant mode or in a PFC mode, but not in both at the same time.

The same watermarks mechanism is used for PFC and for 802.3X FC to determine when to send XOFF and XON packets. When PFC is used in the receive path, priority PAUSE packets are sent instead of 802.3X PAUSE packets. The format of priority PAUSE packets is described in Operation and Rules.

Specific considerations for generating PFC packets:

- When a PFC packet is sent, the packet sets all the UPs that are associated with the relevant TC (UP-to-TC association in receive is defined in RTRUP2TC register).

#### 13.17.2.3.2 Operation and Rules

The *TFCE* field in the Flow Control Configuration (FCCFG) register enables transmission of PAUSE packets as well as selects between the link FC mode and the PFC mode.

The content of the Flow Control Receive Threshold High (FCRTH) register determines at what point the LAN controllers transmit the first PAUSE frame. The LAN controllers monitor the fullness of the receive FIFO and compares it with the contents of FCRTH. When the threshold is reached, the LAN controllers send a PAUSE frame with its pause time field equal to FCTTV.

At this time, the LAN controllers start counting an internal shadow counter (reflecting the pause time-out counter at the partner end). When the counter reaches the value indicated in FCRTV register, then, if the PAUSE condition is still valid (meaning that the buffer fullness is still above the low watermark), an XOFF message is sent again.

Once the receive buffer fullness reaches the low water mark, the LAN controllers send an XON message (a PAUSE frame with a timer value of zero). Software enables this capability with the XONE field of the FCRTL.

The LAN controllers send a PAUSE frame if it has previously sent one and the FIFO overflows. This is intended to minimize the amount of packets dropped if the first PAUSE frame did not reach its target.

#### 13.17.2.3.3 Flow Control High Threshold — FCRTH

The LAN controllers send a PAUSE frame when the Rx packet buffer is full above the high threshold. The threshold should be large enough to overcome the worst case latency from the time that crossing the threshold is sensed until packets are not received from the link partner.

Referring to Annex O of IEEE802.1Qbb rev 2.3, worst case latency depends on three parameters:

1. Maximum frame size over the TC for which FCRTH is computed. It is referred as MaxFrame(TC).
2. Maximum frame size over the link (all TCs altogether). It is referred as MaxFrame(link).
3. Whether or not MACsec is enabled or disabled over the link.

Two values are envisaged for MaxFrame:

- 1.5 KB (Ethernet - jumbo disabled)
- 9.5 KB (jumbo enabled)

Worst case latency, which is referred as Standard Delay Value (Std DV), is given by:

Std DV = MaxFrame(TC) + MaxFrame(link) + PFC Frame + 2 x Cable Delay + 2 x Interface Delay + Higher Layer Delay + (MACSec enabled = yes) x Sec Y Transmit Delay

Std DV (bit time units) = MaxFrame(TC) + MaxFrame(link) + 672 + 2 x 5,556 + 2 x (25,600 + 8,192 + 2 x 2,048) + 6,144 + (MACSec enabled = yes) x (MaxFrame(link) + 3,200)

MaxFrame(TC) term and MaxFrame(link) term included in Sec Y Transmit Delay correspond to worst case scenarios issued by the link partner. All other terms in Std DV formula must take in account worst case incoming traffic pattern which would lead to worst case buffer utilization as per the internal architecture of Rx packet buffers in the LAN controllers.

Internal architecture of the Rx packet buffer has the following restrictions:

1. Any packet starts at 32 byte aligned address.
2. Any packet has an internal status of 32 bytes. As a result, the Rx packet buffer is used at worst conditions when the Rx packet includes 65 bytes that are posted to the host memory. Assuming that the CRC bytes are not posted to host memory then in the worst case the Rx packet buffer can be filled at 1.44 higher rate than the wire speed (69-byte packet including CRC + 8-byte preamble + 12-byte back-to-back IFS consumes 4 x 32 bytes = 128 bytes on the Rx packet buffer).
3. An additional packet from the concerned traffic class may be inserted into the Rx packet buffer due to the internal loopback switch just before it is decided to issue XOFF to the link partner.

FCRTH must be set to the size of the Rx packet buffer allocated to the TC minus the controller DV.

### Table 13-44. LAN Controllers Delay Values (DV) Used For FCRTH

| 9.5 KB Jumbo Enabled | LAN Controllers DV |
|---|---|
| No | 24 KB |
| No | 25 KB |
| Yes | 50 KB |
| Yes | 35 KB |
| No | 27 KB |
| No | 27 KB |
| Yes | 60 KB |
| Yes | 45 KB |

***Note:*** 9.5 KB jumbo enabled/disabled is a global setting per port which concerns all TCs.

#### 13.17.2.3.4 FC Low Threshold — FCRTL

The low threshold value is aimed to protect against wasted available host bandwidth. There is some latency from the time that the low threshold is crossed until the XON frame is sent and packets are received from the link partner. The low threshold must be set high enough so that the Rx packet buffer does not get empty before any new entire packets are received from the link partner. When considering data movement from the Rx packet buffer to host memory, then large packets represent the worst. Assuming the host bandwidth is about the bandwidth on the wire (when dual ports are active at a given time), and assuming a PCIe round trip is required to get the receive descriptors, we get the following formula for FCRTL:

FCRTL = 2 x MaxFrame(TC) + PCIe round trip delay

PCIe round trip delay is assumed to be ~1 μs and it must cover for worst case incoming traffic pattern (buffer utilization by 1.44 than wire rate):

FCRTL (bit time units) = 2 x MaxFrame(TC) + 1.44 x 10,000

Setting the FCRTL to lower values than expressed by the previous equation is permitted. It might simply result with potential sub-optimal use of the PCIe bus once bandwidth is available.

**Table 13-45. LAN Controllers FCRTL**

| 9.5 KB Jumbo Enabled | LAN Controllers DV |
|---|---|
| No | 5 KB |
| No | 7 KB |
| Yes | 21 KB |
| Yes | 7 KB |

#### 13.17.2.3.5 Packet Buffer Size

When FC is enabled, the total size of a TC packet buffer must be large enough for the Low and high thresholds. In order to avoid constant transmission of XOFF and XON frames it is recommended to add some space for hysteresis type of behavior. The difference between the two thresholds is recommended to be at least one frame size (when 9.5 KB jumbo frames are expected over the TC) and larger than a few frames in other cases (4.5 KB for instance). If the available Rx buffer is large enough, it is recommended to increase as much as possible the hysteresis budget. If the available Rx buffer is not large enough it might be required to cut both the low threshold as well as the hysteresis budget.

- For a PFC-enabled TC:
  - FCRTH = FCRTL + hysteresis budget = FCRTL + Max (MaxFrame(TC), 4.5 x 1024 B)
  - Rx packet buffer size = FCRTH + the LAN controller DV (see Section 13.17.2.3.3, "Flow Control High Threshold — FCRTH")
- For a best effort TC:
  - Rx packet buffer size = FCRTL, as the same considerations than described in Section 13.17.2.3.4, "FC Low Threshold — FCRTL" play here to avoid bubbles over PCIe

The total Rx packet buffer size available to a port for all its supported TCs is either 384 KB, 320 KB, or 256 KB, depending on the size allocated to the Flow Director table, 0 KB, 64 KB, or 128 KB, respectively.

The following table assumes four PFC-enabled TCs are defined over the port, of which two are allocated for loss less traffic types like iSCSI, etc. The table lists the recommended settings for the supported combinations. When less than 4 PFC-enabled TCs are defined, and/or when less than 8 TCs are defined, it is recommended to refer to the setting rules described in this section, in Section 13.17.2.3.3, "Flow Control High Threshold — FCRTH", and in Section 13.17.2.3.4, "FC Low Threshold — FCRTL". Note that reducing the number of TCs of a port to what is really needed, helps increasing the port throughput.

**Table 13-46. Some Recommended Rx Packet Buffer Settings**

| Flow Director Table Size | 9.5 KB Jumbo Enabled | Packet Buffer Size of Any of the 4 Best Effort TCs | Packet Buffer Size of Any of The Other 2 PFC-enabled TCs |
|---|---|---|---|
| No | No | 33 KB | 61 KB<br>FCRTL = 6 KB<br>FCRTH = 37 KB |
| No | Yes | 27 KB | 86 KB<br>FCRTL = 21 KB<br>FCRTH = 36 KB |
| No | No | 32 KB | 63 KB<br>FCRTL = 6 KB<br>FCRTH = 36 KB |
| No | Yes | 22 KB | 91 KB<br>FCRTL = 21 KB<br>FCRTH = 31 KB |
| 64 KB | No | 25 KB | 53 KB<br>FCRTL = 6 KB<br>FCRTH = 29 KB |
| 64 KB | Yes | 19 KB | 78 KB<br>FCRTL = 20 KB<br>FCRTH = 28 KB |
| 64 KB | No | 24 KB | 55 KB<br>FCRTL = 6 KB<br>FCRTH = 28 KB |
| 64 KB | Yes | 14 KB | 83 KB<br>FCRTL = 18 KB<br>FCRTH = 23 KB |
| 128 KB | No | 17 KB | 45 KB<br>FCRTL = 6 KB<br>FCRTH = 21 KB |
| 128 KB | No | 16 KB | 47 KB<br>FCRTL = 6 KB<br>FCRTH = 20 KB |

*Note:* In some of the previous cases, it has been necessary to get compromised on the rules for hysteresis and FCRTL in order to fit the size available for Rx packet buffer.

*Note:* In some other cases, after having applied all the rules there was an exceeding available Rx packet buffer left that has been used to extend the hysteresis budgets.

*Note:* In all cases, FCRTH rule has been applied as is, since compromising on it is not allowed and extending it provides no performance benefits.

### 13.17.3 Inter Packet Gap (IPG) Control and Pacing

The LAN controllers support transmission pacing by extending the IPG (the gap between consecutive packets). The pacing mode enables the average data rate to be slowed in systems that cannot support the full link rate (10 GbE or 1 GbE). As listed in Table 13-47, "Pacing Speeds at 10 GbE Link Speed", the pacing modes work by stretching the IPG in proportion to the data sent. In this case the data sent is measured from the end of preamble to the last byte of the packet. No allowance is made for the preamble or default IPG when using pacing mode.

**Example 13-1.**

Consider a 64-byte frame. To achieve a 1 GbE data rate when link rate is LAN and packet length is 64 bytes (16 Dwords), add an additional IPG of 144 Dwords (nine times the packet size to reach 1 GbE). When added to the default IPG gives an IPG of 147 Dwords.

**Example 13-2.**

Consider a 65-byte frame. To achieve a 1 GbE data rate when link rate is LAN and packet length is 65 bytes (17 Dwords when rounded up) add an additional IPG of 153 Dwords (nine times the packet duration in Dwords). When added to the default IPG gives an IPG of 156 Dwords. Note that in this case, where the packet length counted in Dwords is not an integer, count any fraction of a Dword as an entire Dword for computing the additional IPG.

Table 13-47, "Pacing Speeds at 10 GbE Link Speed" lists the pacing configurations supported by the SoC LAN controller at link rates of 10 GbE. When operating at lower link speeds the pacing speed is proportional to the link speed.

**Table 13-47. Pacing Speeds at 10 GbE Link Speed**

| Pacing Speeds (Gb/s) | Delay Inserted into IPG | Register Value |
|---|---|---|
| 10 (10 GbE) | None | 0000b |
| 9.294196 (WAN) | 1 byte for 13 transmitted | 1111b |
| 9.0 | 1 Dword for 9 transmitted | 1001b |
| 8.0 | 1 Dword for 4 transmitted | 1000b |
| 7.0 | 3 Dwords for 7 transmitted | 0111b |
| 6.0 | 2 Dwords for 3 transmitted | 0110b |
| 5.0 | 1 Dwords for 1 transmitted | 0101b |
| 4.0 | 3 Dwords for 2 transmitted | 0100b |
| 3.0 | 7 Dwords for 3 transmitted | 0011b |
| 2.0 | 4 Dwords for 1 transmitted | 0010b |
| 1.0 | 9 Dwords for 1 transmitted | 0001b |
| 10 | None | Default |

***Note:*** Pacing is configured in the *PACE* field of the Pause and Pace (PAP) register.

## 13.18 Physical Functions (PF), Virtual Functions (VF) and ARI (Alternative RID Interpretation)

ATS and ARI are required to support PF and VF. Both ATS and ARI are required in order for PCIe configuration space to be able to address higher than 7 VFs PCIe configurations traditionally configures PF and VFs using 16 bits. ATS and ARI allow more (higher) than 7 functions to be addressed.

The number of VFs, in addition to the PFs, are indicated in the system shared SPI Flash section for the LAN controllers LAN0 and LAN1. A function in the shared SPI Flash image known as I/O Virtualization (IOV) is also need to be enabled. These newly created VF shows up in PCIe configuration space as base/device/function.

In short, to enable hardware based Single Root I/O Virtualization (SR-IOV) and benefit from use of VFs and the built-in Virtual Ethernet Bridge (VEB), you must have:

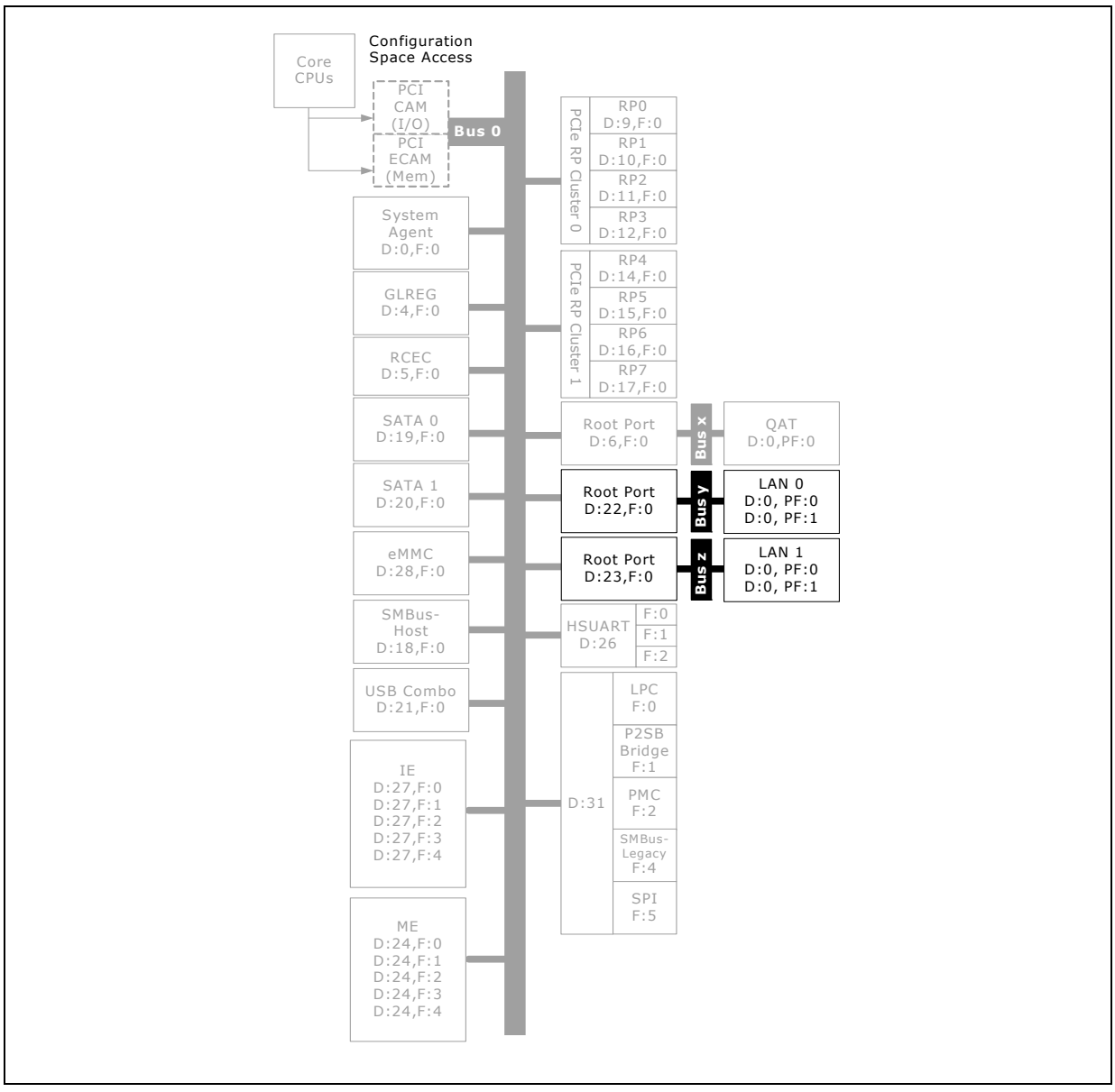
1. Ensure that ATS/ARI are supported in your version of BIOS.
2. Enable SR-IOV in your BIOS.
3. Use an OS that supports virtualization.
4. Enable IOV in the NVM section and indicated the # of VMs desired to create.

For further information regarding SR-IOV, refer to the published SR-IOV Primer titled *PCI-SIG* SR-IOV Primer: An Introduction to SR-IOV Technology*.

## 13.19 Register Map

The software-accessible registers are outlined in Table 13-12, "Base Address Registers" and are depicted here.
Figure 13-14, "Register Map" shows the associated registers from a system viewpoint.

**Figure 13-14.Register Map**




§ §

# 14 USB Combo Controller

This chapter describes the Universal Serial Bus (USB) Combo Controller. It provides USB 2.0 and USB 3.0 ports.

**Figure 14-1. What is Covered in This Chapter**

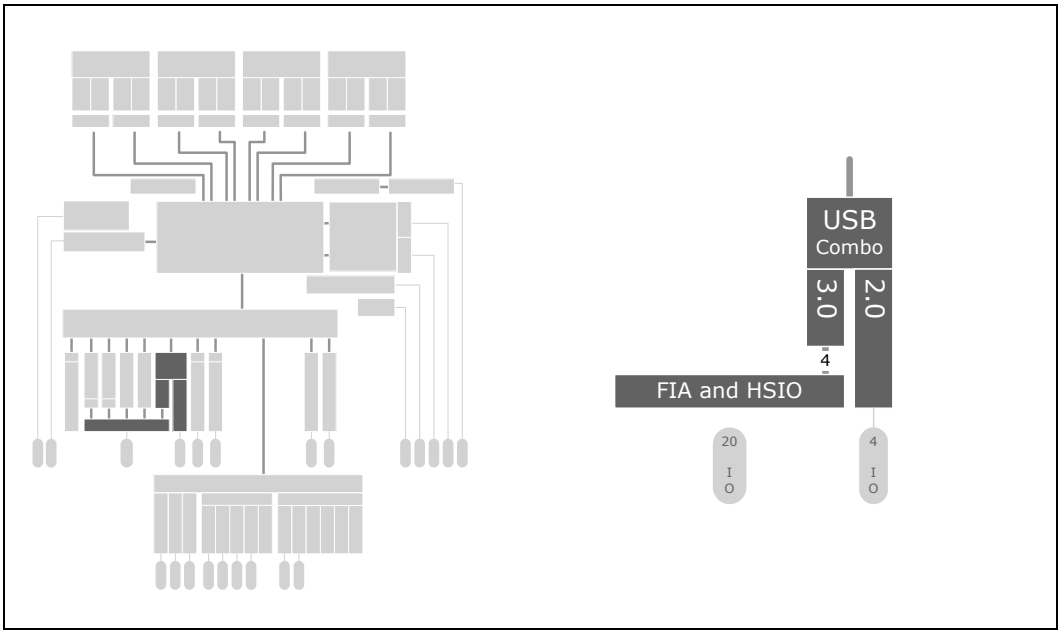



**Table 14-1. References**

| Document Title | Document ID / Location |
|---|---|
| *Universal Serial Bus 3.0 Specification, Revision 1.0* | http://www.usb.org/developers/docs/ |
| *Universal Serial Bus Specification, Revision 2.0* | http://www.usb.org/developers/docs/usb20_docs/ |
| *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1* | https://www.intel.com/content/www/us/en/io/universal-serial-bus/extensible-host-controler-interface-usb-xhci.html |
| *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* | https://www.intel.com/content/www/us/en/io/universal-serial-bus/ehci-specification-for-usb.html |

## 14.1 Signal Descriptions

The signal descriptions are shown in Table 14-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. Also see Table 39-3.The Direction column of Table 39-3 on page 1028 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- Differential = Differential signal pair. Two pins: Positive; Negative.

**Table 14-2. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| USB3_TX_DP[3:0]<br>USB3_TX_DN[3:0] | O, Differential | Yes | **USB 3.0 Transmit Data:** Differential output signals from the USB Combo Controller to the FIA/PHY. |
| USB3_RX_DP[3:0]<br>USB3_RX_DN[3:0] | I, Differential | Yes | **USB 3.0 Receive Data:** Differential input signals to the USB Combo Controller from the FIA/PHY. |
| USB2_DP[3:0]<br>USB2_DN[3:0] | I,O, Differential | No | **USB 2.0 Transceiver Data:** Differential bi-directional signal pins. |
| USB_OC_N | I | No | **USB Over Current.** Active low input signal from the platform board that alerts the SoC that a VBUS over-current condition exists. |
| USB2_COMP | I | No | **Compensation:** The platform board must provide a 113 Ω, 1% resistor connected from this pin to VSS. |

## 14.2 Feature List

The SoC supports up to four USB 3.0 compliant ports depending on product SKU.

The SoC provides these ports through a USB 3/USB 2 Combo Controller which is an integrated PCI Express* function, Bus 0, Device 21 (decimal), Function 0.

All ports operate in Host Mode. The USB 2.0 operation supports USB 1.X devices and a Rate Matching Hub (RMH) is provided for USB 1.1 devices.

The controller contains one eXtensible Host Controller Interface (xHCI) with software, which complies to the *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1*.

*Note:* For a USB 3.0-compliant port design, one set of the controller USB 3.0 signals and one set of the USB 2.0 signals must be routed to the same port connector. Thus for USB 3.0 specification compliance, the USB Combo Controller (xHCI) can only support four devices.

*Note:* USB On-The-Go (OTG) not supported.

### 14.2.1 USB 3.0

The USB 3.0 Super Speed data interface is dual-simplex, four-wire differential signals (2-TX and 2-RX) which are separate from the USB 2.0 signals and supports simultaneous bi-directional (full Duplex) data flows. The interface supports a bit rate of 5 Gbps with a maximum data throughput of less than 4 Gbps due to usage of an 8b/10b symbol encoding scheme and additional protocol overhead.

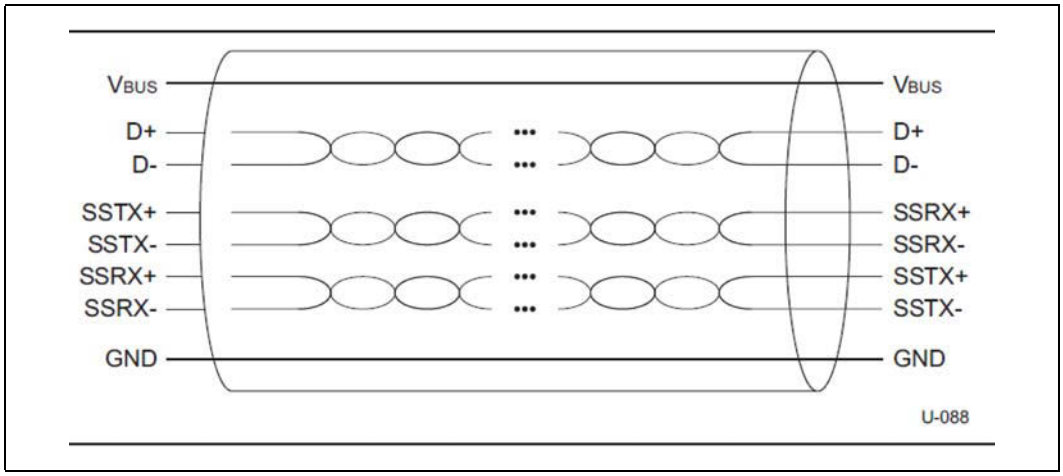
### 14.2.2 USB 2.0

The USB 2.0 data interface is half-duplex two-wire differential signals, which support unidirectional data flow with negotiated directional bus transitions. The USB 2.0 supports bit rates of:

- Low-speed (1.5 Mbps)
- Full-speed (12 Mbps)
- High-speed (480 Mbps)

For additional information refer to the *Universal Serial Bus Specification, Revision 2.0*.

**Figure 14-2. USB 3.0 Cable**



### 14.2.3 xHCI Interface

The USB 3.0 xHCI Host Controller (xHCI) supports the following features:

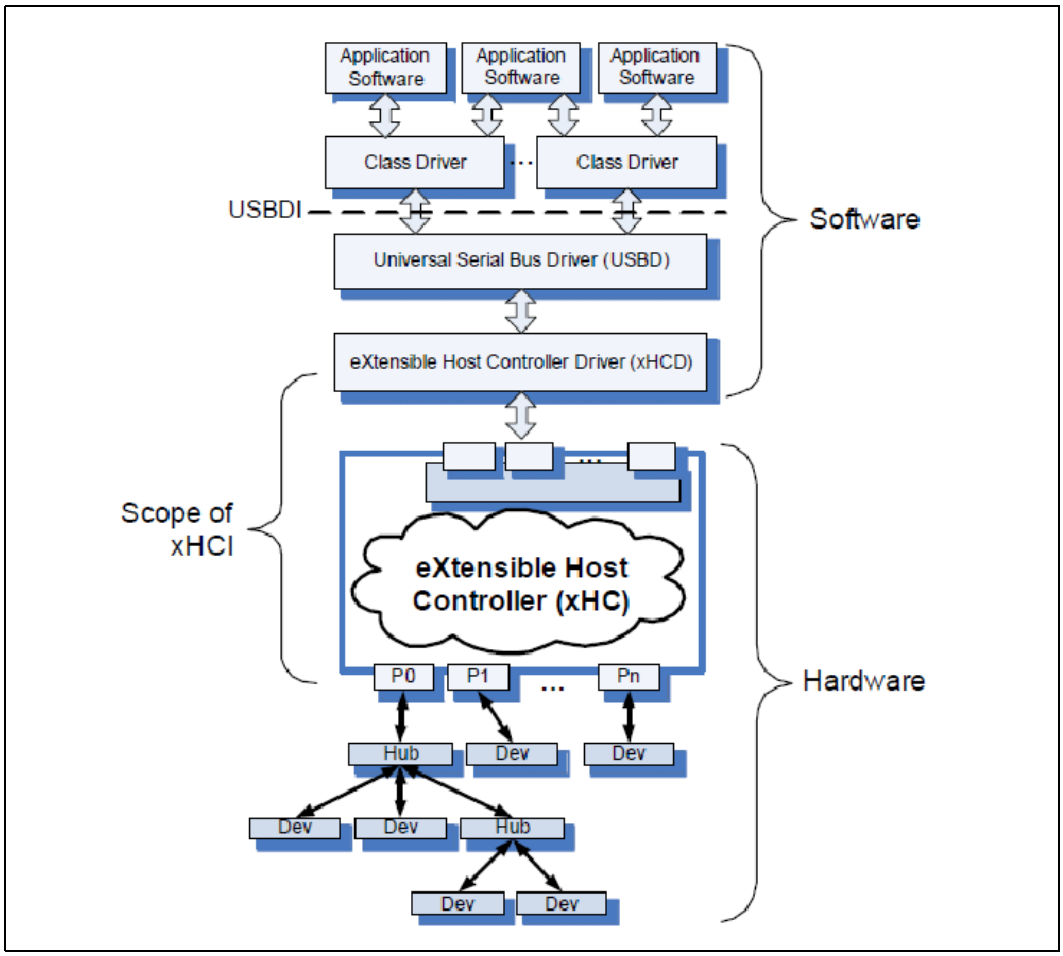
- Compliant with the specification for USB 1.x, 2.0 (1.5 Mbps, 12 Mbps, 480 Mbps) and USB 3.0 Super Speed (5 Gbps). Supports all USB transactions (bulk, isochronous, interrupt, control).
- USB ports support up to 32 slots and 96 end-points
- Supports xHCI software host controller interface
- Supports wakeup from suspend states S1-S5 and remote-suspend wakeup
- Per port USB disable
- Support prefetch-based pause feature
- Support of USB suspend mode including PLL turn off in S0/Sx
- Capability to use reduced Frame List Sizes
- 64-bit data and 36-bit address
- Downstream 1DW memory read and writes, upstream memory read and writes
- Intel® Direct Connect Interface (Intel® DCI) customer visible debug port
- Intel® Direct Connect Interface (Intel® DCI) over DbC
- Intel® Direct Connect Interface (Intel® DCI) over Intel® DCI OOB (Out of Band)
- xHCI Debug Capability
- EP Type based Port Lock Capability
- The following USB capabilities are not supported:
  - Port Disable Override capability
  - USB-R Capability

## 14.3 Architectural Overview

The following section is adopted from *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1*.

Table 14-3, "Universal Serial Bus, Revision 3.0 System Block Diagram" shows a conceptual block diagram of the building block layers in a USB host system that work in concert to support USB 3.0

**Figure 14-3. Universal Serial Bus, Revision 3.0 System Block Diagram**



The component layers are:

- **Application Software.** This software uses the services provided by one or more USB devices. Application software interfaces with USB devices through standardized interfaces provided by the Class Drivers.
- **Class Driver Software.** This software executes on the host PC corresponding to a particular "class" of USB device (Mass Storage, Human Interface, Audio, etc.). Class Driver software is typically part of the operating system or provided with the USB device.
- **USB Driver (USBD).** The USBD is a system software Bus Driver that abstracts the details of the particular Host Controller Driver for a particular operating system. The generic USB interface presented to the system by USBD is referred to as the *USB Driver Interface* or the **USBDI**.
- **eXtensible Host Controller Driver (xHCD).** xHCD provides the software layer between the Host Controller hardware and the USBD. The details of the host controller driver depend on the host controller hardware register interface definition.
- **eXtensible Host Controller (xHC).** The host controller is the specific hardware implementation of the host controller architecture. There is one host controller specification for the USB 3.0 host controller, which enables support for Low-, Full-, High- and SuperSpeed devices. The interface presented by the xHC to the system is referred to as the *eXtensible Host Controller Interface* or the **xHCI**.
- **USB Device.** This is a hardware device that expands the bus topology (hub) or performs a useful end-user function. Interactions with USB devices flow from the applications through the software and hardware layers to the USB devices.

A key feature of the USB architecture is the **Device Framework** that it presents. The Device Framework defines the interface between a USB device and a Class Driver, which is independent of the particular host controller interface that a system employs to communicate with the USB. This interface consists of a Default Pipe, and zero or more additional class defined Pipes. The Default Pipe (also referred to as the *Default Control Endpoint*) is used to enumerate and manage a USB device. It can also be used to provide access to application specific features of the device. The class defined Pipes provide specialized Quality-of-Service requirements to perform device class specific functions.

The Device Framework allows the USB architecture to separate the details of the "Bus" interface from that of the application specific ("Device") interface, resulting in a split driver model (xHCD/Class Driver). Note that in this context, Device Class refers to the portion of a USB device that performs some useful end user application specific function (e.g. Mass Storage, Audio, Human Interface, etc.).

The USB bus driver (USBD) provides a standard method of interfacing to the transport mechanisms (USB Framework) defined by the USB architecture (Isoch, Interrupt, Control, and Bulk Pipes) and the Device Class driver is where all the application specific knowledge resides. A Class Driver will also include any "value add" that a vendor may provide. As long as the USB Framework presented through the USBDI remains unchanged, the USB Class Drivers do not have to change because the USB bus driver does (e.g. to support the xHCI).

Working groups in the USB-IF have defined several standard USB Device Classes (Mass Storage, Audio, etc.). A USB device vendor may choose to define a proprietary Device Class for their product or utilize part or all of an appropriate USB-IF defined Device Class. The USB-IF defined Device Classes provide a baseline set of features, for their respective class. Several USB Device Classes are supported natively by today's Operating Systems.

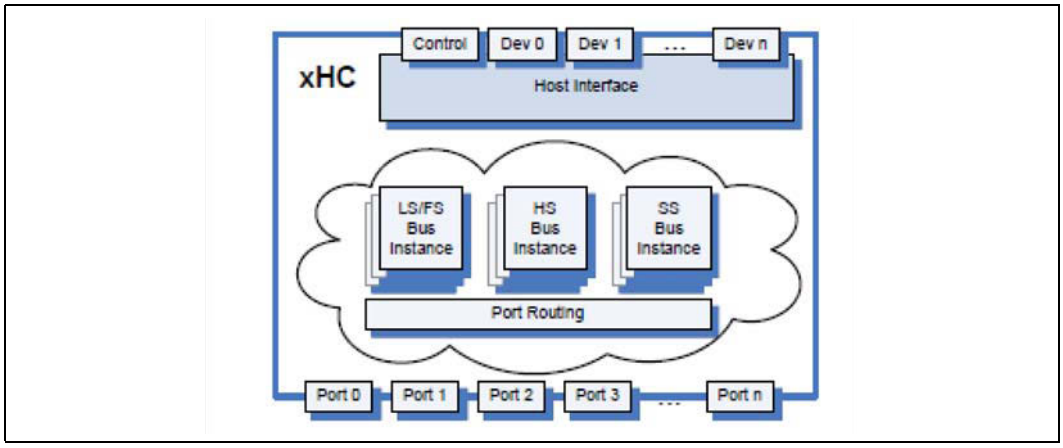
Native OS support for Device Classes allows a compliant device to provide a user with basic functionality if the vendor Device Class drivers are not available, however a vendor can define their own Class Driver to add value. Many commodity USB device vendors (mice, keyboard, etc.) take advantage of those provided by OS vendors and don't bother to offer their own Class Drivers. If a vendor offers a USB device that does not fall under one of the standard USB defined Device Classes supported by an OS then they shall offer their own Class Driver.

The xHCI is used for all communications to devices connected through the Root Hub ports of the USB 3.0 host controller.

The xHCI architecture allows the USB 3.0 host controller to provide USB functionality for all speed devices without requiring, as in previous generations, companion controllers along with the associated software support for their respective drivers. The enhanced features of the xHCI architecture are key to delivering this simplified operating environment.

Note that the Figure below does not imply a particular xHC implementation, however the functional partitioning that it illustrates is useful for this discussion. The Host Interface Logic manages the Registers and DMA associated with the xHC.

**Figure 14-4. USB 3.0 eXtensible Host Controller**




The xHC always manages the respective speed USB devices connected to its Root Hub ports. Depending on the implementation, the resources of a USB bus instance (bandwidth, device addressability, etc.) may be presented on each root hub port, shared across multiple root hub ports, or a combination of allocations.

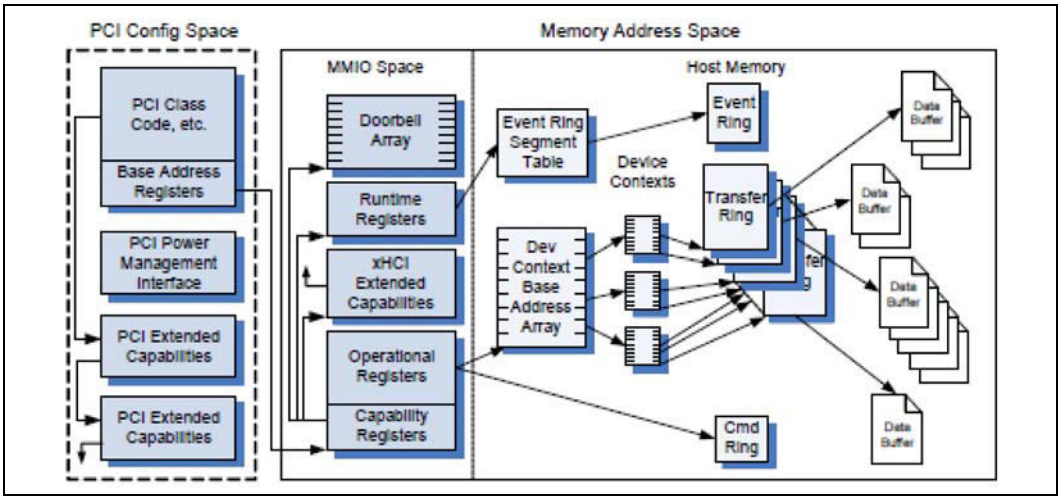
This specification defines the registers and interfaces for the eXtensible Host Controller Interface.

### 14.3.1 xHCI Interface Architecture

The xHCI interface defines three interface spaces (refer to in the Figure below):

- **Host Configuration Space.** Every xHC implementation shall include a means of identifying and enumerating the host controller by system software. This specification provides a PCI example of the Host Configuration Space, which is referred to as **PCI Config Space**. The PCI Config Space definition provides a working example of configuration space use for system xHC enumeration and resource (interrupt, power, virtualization, etc.) management.
- **MMIO Space.** The Register Space represents the hardware registers presented by the xHC to system software that reside in the Memory Address Space. The Register Space provides for the implementation-specific parameters defined in the xHCI normal and Extended Capabilities registers, the Operational and Runtime control and status registers, and the Doorbell Array used to flag accesses to individual USB devices. This space, normally referred to as I/O space, is implemented as Memory-Mapped I/O (**MMIO**) space.
- **Host Memory.** This space is defined by the control data structures (Device Context Base Address Array, Device Contexts, Transfer Rings, etc.) and data buffers that are allocated and managed by the xHC Driver to enable the endpoint traffic of individual devices. This space is allocated in the Kernel and User areas of the Memory Address Space.

**Figure 14-5. General Architecture of the eXtensible Host Controller Interface**




The xHCI provides support for two categories of USB transfer types: asynchronous and periodic. Isochronous and Interrupt transfers are Periodic transfer types. Asynchronous transfer types include Control and Bulk. Figure 3 illustrates that the xHCI provides a homogeneous mechanism (Transfer Rings) for each category of transfer type.

The USB Base Address Register (BAR) in the PCI Config Space points to the base address of the xHC register interface. The xHC register interface consists of 4 major components: Capability Registers, Operational Registers, Runtime Registers, and the Doorbell Array. The Operational and Capability Registers are concatenated in MMIO space. The Runtime Registers are actually just an extension of the Operational Registers. Their partitioning allows the xHC to better support virtualization, by allowing the Runtime Registers to reside on a separate page boundary. A xHCI Capabilities Pointer mechanism (similar to that defined by PCI) is presented in the Capability Registers to point to new or optional capabilities of an xHC implementation.

The ***Capability Registers*** specify read-only limits, restrictions and capabilities of the host controller implementation. These values are used as parameters to the host controller driver.

The ***Runtime*** and ***Operational Registers*** specify host controller configuration and runtime modifiable state, and are used by system software to control and monitor the operational state of the host controller. These registers are partitioned as a function of those that are heavily accessed during runtime and those that are accessed only at initialization time or only lightly during runtime to better support virtualization of the xHCI.

The ***xHCI Extended Capabilities*** specify optional features of an xHC implementation, as well as providing the ability to add new capabilities to implementations after the publication of this specification.

The ***Doorbell Array*** is an array of up to 256 Doorbell Registers, which supports up to 255 USB devices or hubs. Each ***Doorbell Register*** provides system software with a mechanism for notifying the xHC if it has Slot or Endpoint related work to perform. A *DB Target* field in the Doorbell Register is written with a value that identifies the reason for **"ringing"** the doorbell. Doorbell Register 0 is allocated to the Host Controller for Command Ring management.

The term ***Device Slot*** is used as a generic reference to a set of xHCI data structures associated with an individual USB device. Each device is represented by an entry in the *Device Context Base Address Array*, a register in the *Doorbell Array* register, and a device's *Device Context*. The term ***Slot ID*** refers to the index used to identify a specific *Device Slot*. For example the value of *Slot ID* will be used as an index to identify a specific entry in the *Device Context Base Address Array*.

The ***Device Context Base Address Array*** supports up to 255 USB devices or hubs, where each element in the array is a pointer to a *Device Context* data structure.

The **Command Ring** is used by software to pass device and host controller related commands to the xHC. The *Command Ring* shall be treated as read-only by the xHC. Refer to section 4.9.3 for a discussion of Command Ring Management.

The **Event Ring** is used by the xHC to pass command completion and asynchronous events to software. The *Event Ring* shall be treated as read-only by system software. Refer to section 4.9.4 for a discussion of Event Ring Management.

A ***Transfer Ring*** is used by software to schedule work items for a single USB Endpoint. A Transfer Ring is organized as a circular queue of ***Transfer Descriptor*** (TD) data structures, where each *Transfer Descriptor* defines one or more Data Buffers that will be moved to or from the USB. *Transfer Rings* are treated as read-only by the xHC. Refer to section 4.9.2 for a discussion of Transfer Ring Management.

All three types of rings support the ability for system software to grow or shrink them while they are active. Special TDs written to the Transfer and Command rings allow software to change their size, however since the Event Ring is read-only to software, the **Event Ring Segment Table** is provided so that software may modify its size.

## 14.4 xHCI Data Structures

The xHC is expected to run in virtual memory environments where the size of a contiguous block of physical memory will be limited by the Page size of the system. The data structures that the xHC uses to manage devices and endpoints are designed to accommodate this limitation, by either keeping the data structure under 4K Bytes (the minimum Page size supported), or providing mechanisms to link non-contiguous blocks of physical memory to form larger, logically contiguous data structures, e.g. circular queues of data structures that point to the data buffers used for transferring USB data to or from the host. The data buffers referenced by these data structures may be byte aligned and reference from 1 to 64K bytes of contiguous physical data.

Please refer to the xHCI Specification for further information.

## 14.5 Command Interface

To manage the xHC and the devices attached to it, the xHC provides an independent Command Ring interface. A work item on a Command Ring is called a *Command Descriptor* (CD). Command Ring operation is very similar to that of Transfer Rings, software issues a command to the xHC by placing a CD on the Command Ring then rings the Host Controller doorbell. The size of the Command Ring can be modified using the same Link TRB mechanism that Transfer Rings use. All commands result in a *Command Completion Event* being placed on the Event Ring, which reports the completion status of the command. Commands are executed by the xHC in the order that they are placed on the Command Ring. System software may add CDs to the Command Ring while it is running, however the execution of CDs should be stopped if software wants to delete or reorder (i.e. raise the priority of) scheduled CDs. Special Command Ring controls allow commands to be stopped or aborted.

Please refer to the *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* for further information.

## 14.6 Root Hub Management

The host controller of a USB bus is required to implement Root Hub functionality. The Operational Register space contains port registers that provide the hardware status and control needed to manage each port within the USB Specification. An xHC Root Hub may provide USB 2.0 and USB 3.0 Root hub ports to support Low-, Full-, or High-Speed as well as SuperSpeed devices. The host controller traverses the Transfer Rings and encounters work items that result in the host controller executing USB transactions. These transactions are routed to the Root Hub port associated with the attached downstream USB device. The port registers provide system software with the control and status information required to manipulate the port in accordance with the USB Specification. The supported features include: detecting device connects, disconnects, performing device resets, manipulating port power and managing port power management capabilities. System software should provide an abstraction to the USB system software stack that allows the Root Hub ports to be manipulated by the system as if they were ports on an external hub.

## 14.7 Operational Model

Please refer to the xHCI Specification Chapter 4 “Operational Model” for information related to the eXtensible Host Controller Interface (xHCI) hardware and eXtensible Host Controller Driver (xHCD) (generally referred to as system software).

The chapter includes significant operational feature of the eXtensible Host Controller (xHC) and operational model requirements for the xHC hardware.

## 14.8 Supported Transaction Types

The controller is discovered as bus 0, device 21, function 0.

Table 14-3, "Transaction Type Supported by USB Block" shows the transaction types supported by the USB block:

### Table 14-3. Transaction Type Supported by USB Block

| Transaction Type | Upstream | Downstream |
|---|---|---|
| Memory Read (32/64 bits) | Yes | Yes |
| Memory Write (32/64 bits) | Yes | Yes |
| I/O Read | No | No |
| I/O Write | No | No |
| Configuration Read0 | No | Yes |
| Configuration Write0 | No | Yes |
| Configuration Read1 | No | No |
| Configuration Write1 | No | No |
| Message | No | No |
| MessageD | No | No |
| Completion | Yes | Yes |
| CompletionD | Yes | Yes |

## 14.9 Reset Requirements

Table 14-4, "Reset Signal Information" shows the reset signal information used in the USB Controller.

### Table 14-4. Reset Signal Information

| Reset Signal | Power Well | Assertion | Description |
|---|---|---|---|
| AON Power Good Based Reset | SUS | Async Power Good Removal | Power Good based reset to be used for isolation |
| Reset | SUS | Async SoC Controlled conditions | Internally synchronous to SUS clock, after AON power is stable and reset sequence is complete. |
| VNN Power Good Based Reset | VNN/CORE | Async Power Good Removal | Power Good based reset to be used for isolation |
| SBI Reset | VNN/CORE | Async Fabric Reset | Internally Synchronous to Side Clock. |
| Fabric Reset | VNN/CORE | Async Host Partition Reset | Internally Synchronous to backbone clock. |

## 14.10 USB Power Management – Link States

Table 14-5, "USB 2.0 Link States and Power Actions" explains the USB 2.0 link states and power actions. Each deeper link state includes actions from prior/shallower link states unless explicitly stated otherwise. So details are not repeated in each deeper link state.

### Table 14-5. USB 2.0 Link States and Power Actions

| Link State | Entry | Exit | Core Action | PHY Action |
|---|---|---|---|---|
| **USB2** | | | | |
| U0 Idle | All scheduled transactions completed for BI.<br>Minimum time left before next BI. | Start of new BI (early indication used to wake PLL). | Fabric Clock allowed to be trunkgated if other ports do not need it.<br>Fabric PLL allowed to be shut down if no other port needs it.<br>All RFpower gating. | x and Rx current sources and Bias circuits are actively managed.<br>Tx shut down if Receiving.<br>Rx shut down if Transmitting. |
| U2 | L1 is enabled.<br>Device on root port supports L1.<br>Device on root port accepts L1 entry required. | Host DB ring.<br>Device initiates exit from L1.<br>Optional timer based wake for non-compliant devices. | Update LTR.<br>Clock gate USB2 link clock at port level.<br>Shut down USB2 PLL if all ports are in L1 or deeper.<br>Indicate "suspend" to PHY. | Shut down HS clocks.<br>Power gate core portion of PHY. |
| U3 | SW places port in U3. | SW places port in U0. | Update LTR to "no requirement".<br>Allow D3 based actions (implicit).<br>Indicate L1 "suspend" to PHY. | Power gate core portion of PHY. |
| Disconnected | | | Update LTR to "no requirement".<br>Allow D3 based actions (implicit).<br>Indicate L1 "suspend" to PHY. | Power gate core portion of PHY. |

Table 14-6, "USB 3.0 Link States and Power Actions" explains the USB 3.0 link states and power actions. Each deeper link state includes actions from prior/shallower link states unless explicitly stated otherwise. So details are not repeated in each deeper link state.

**Table 14-6. USB 3.0 Link States and Power Actions**

| Link State | Entry | Exit | Core Action | PHY Action |
|---|---|---|---|---|
| **USB3** | | | | |
| U0 | | | Update LTR based on received BELT messages | |
| U0 Idle | All scheduled transactions completed for BI.<br>Minimum time left before next BI. | Start of new BI (early indication used to wake PLL). | Fabric Clock allowed to be trunkgated if other ports do not need it.<br>Fabric PLL allowed to be shut down if no other port needs it.<br>All RFpower gating. | None. |
| U1 | U1 idle time expires. | DB ring or Periodic transaction scheduled.<br>Device initiated exit.<br>U2 timer expires. | Link clocks gated at port level.<br>PIPE state = PS1. | Tx OFF.<br>Rx OFF.<br>Maintain common mode. |
| U2 | U2 idle timer expires. | DB ring or Periodic transaction scheduled.<br>Device initiated exit. | Link clocks gated at port level.<br>PIPE state = PS3.<br>Indicate USB 3.0 link PLL can be shut off.<br>Indicate Morhy port can be power gated.<br>Aux clock gated at the partition level.<br>Requires: Timer tick for RxDetect timers. | PLL off.<br>Tx, Rx off.<br>Common mode off.<br>Power gate core well. |
| U3 | SW initiated. | SW initiated. | Allow D3 actions similar to U2. | Similar to U2. |
| RxDetect. | NA | NA | PIPE state = PS3 (after first RxDetect).<br>Similar to U3. | Similar to U3. |

## 14.11 Error Condition and Handling

The USB 3.0 port Error handling is as per USB3 and xHCI Specifications.

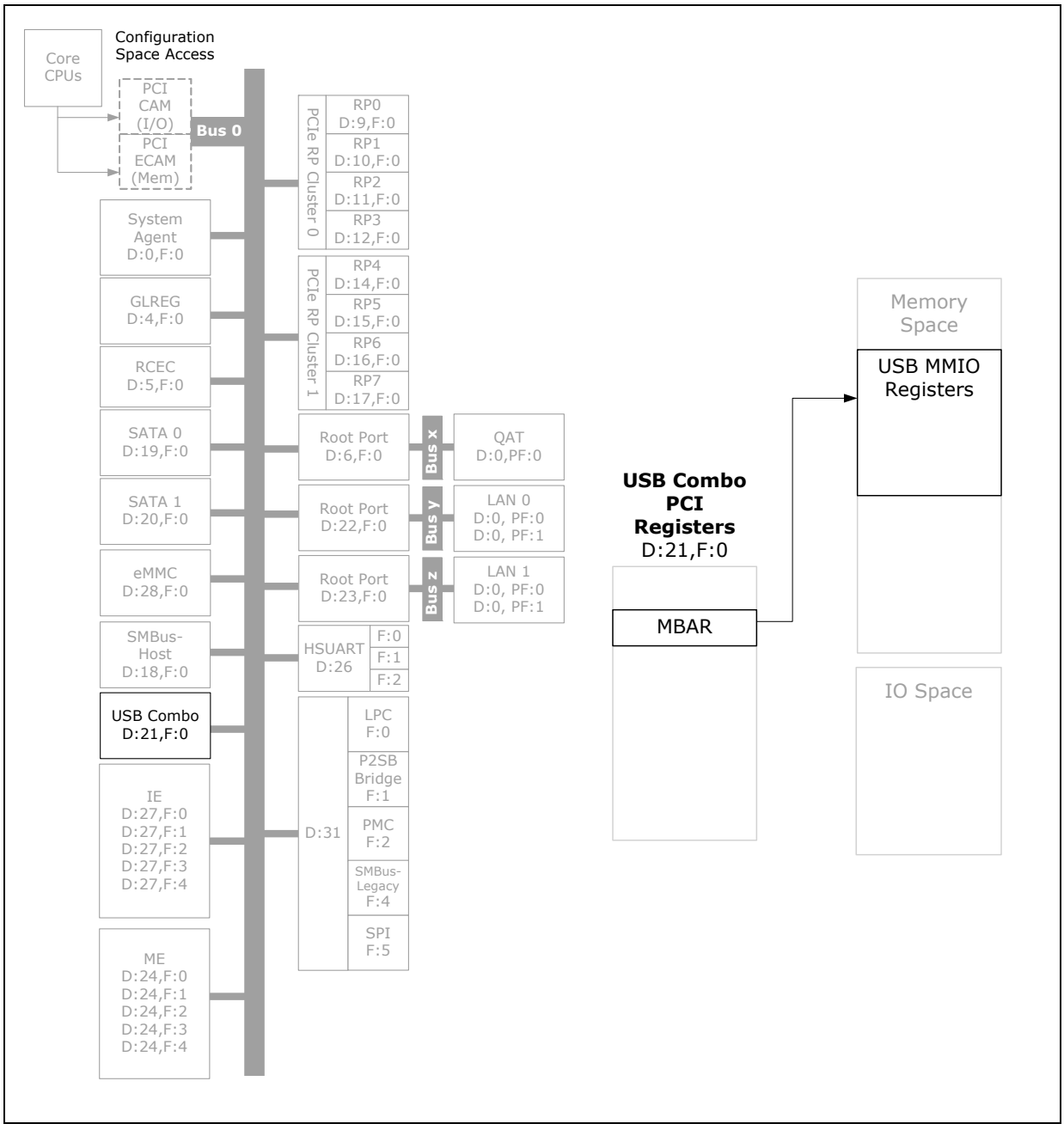
## 14.12 Debug Capability (DbC)

The USB Debug Capability (DbC) is provided by the SoC USB eXtensible Host Controller (xHC). DbC is an optional USB functionality that enables low-level system debug over USB. The xHC Interface (xHCI) debugging capability provides a means of connecting two systems where one system is a Debug Host and the other a Debug Target (System Under Test). See the section on DbC in the *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1*. Also see Section 38.4.1, "Debug Capability (DbC)" on page 988, in this document.

## 14.13 Register Map

Figure 14-6 shows the associated registers from a system software viewpoint.

**Figure 14-6. Register Map**



## 14.13.1 PCI Configuration and Capabilities

### Table 14-7. Configuration and Capabilities Register Map (Sheet 1 of 2)

| Offset from Configuration Space of B0:D21:F0 (hex) | Register Short Name | Description |
|---|---|---|
| **Standard Type 0 PCI Configuration Registers** | | |
| 0 | VID | Vendor ID |
| 2 | DID | Device ID |
| 4 | CMD | Command |
| 6 | STS | Device Status |
| 8 | RID | Revision ID |
| 9 | PI | Programming Interface |
| A | SCC | Sub Class Code |
| B | BCC | Base Class Code |
| D | MLT | Master Latency Timer |
| E | HT | Header Type |
| 10 | MBAR | Memory Base Address |
| 2C | SSVID | USB Subsystem Vendor ID |
| 2E | SSID | USB Subsystem ID |
| 34 | CAP_PTR | Capabilities Pointer |
| 3C | ILINE | Interrupt Line |
| 3D | IPIN | Interrupt Pin |
| **PCI Power Management Capability** | | |
| 70 | PM_CID | PCI Power Management Capability ID |
| 71 | PM_NEXT | Next Item Pointer #1 |
| 72 | PM_CAP | Power Management Capabilities |
| 74 | PM_CS | Power Management Control/Status |
| **Message Signaled Interrupt Capability** | | |
| 80 | MSI_CID | Message Signaled Interrupt CID |
| 81 | MSI_NEXT | Next item pointer |
| 82 | MSI_MCTL | Message Signaled Interrupt Message Control |
| 84 | MSI_MAD | Message Signaled Interrupt Message Address |
| 88 | MSI_MUAD | Message Signaled Interrupt Upper Address |
| 8C | MSI_MD | Message Signaled Interrupt Message Data |

## Table 14-7. Configuration and Capabilities Register Map (Sheet 2 of 2)

| Offset from Configuration Space of B0:D21:F0 (hex) | Register Short Name | Description |
|---|---|---|
| **Device-Specific PCI Configuration Registers** | | |
| 40 | XHCC1 | XHC System Bus Configuration 1 |
| 44 | XHCC2 | XHC System Bus Configuration 2 |
| 50 | XHCLKGTEN | Clock Gatings |
| 58 | AUDSYNC | Audio Time Synchronization |
| 60 | SBRN | Serial Bus Release Number |
| 61 | FLADJ | Frame Length Adjustment |
| 62 | BESL | Best Effort Service Latency |
| 90 | DEVIDLE | Device Idle Capability |
| 94 | VSHDR | Vendor Specific Header |
| 98 | SWLTRPTR | SW LTR POINTER |
| 9C | DEVIDLEPTR | Device Idle Pointer Register |
| A0 | DEVIDLEPOL | Device Idle Power ON Latency |
| A4 | HSCFG2 | High Speed Configuration 2 |
| B0 | U2OCM1 | XHCI USB2 Overcurrent Pin Mapping 1 |
| D0 | U3OCM1 | XHCI USB3 Overcurrent Pin Mapping 1 |
| FC | XHCC3 | XHCC3 |

## 14.13.2 Memory-Mapped Registers

### Table 14-8. Relocatable Memory-Mapped Registers (Sheet 1 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **eXtensible Host Controller (xHCI) Capability Registers** | | | |
| MBAR | 0 | CAPLENGTH | Capability Registers Length |
| MBAR | 2 | HCIVERSION | Host Controller Interface Version Number |
| MBAR | 4 | HCSPARAMS1 | Structural Parameters 1 |
| MBAR | 8 | HCSPARAMS2 | Structural Parameters 2 |
| MBAR | C | HCSPARAMS3 | Structural Parameters 3 |
| MBAR | 10 | HCCPARAMS | Capability Parameters |
| MBAR | 14 | DBOFF | Doorbell Offset |
| MBAR | 18 | RTSOFF | Runtime Register Space Offset |
| **xHCI Operational Registers** | | | |
| MBAR | 80 | USBCMD | USB Command |
| MBAR | 84 | USBSTS | USB Status |
| MBAR | 88 | PAGESIZE | Page Size |
| MBAR | 94 | DNCTRL | Device Notification Control |
| MBAR | 98 | CRCR_LO | Command Ring Low |
| MBAR | 9C | CRCR_HI | Command Ring High |
| MBAR | B0 | DCBAAP_LO | Device Context Base Address Array Pointer Low |
| MBAR | B4 | DCBAAP_HI | Device Context Base Address Array Pointer High |
| MBAR | B8 | CONFIG | Configure |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 2 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **xHCI USB Port Registers** | | | |
| MBAR | 480 | PORTSC1 | Port Status and Control USB2 |
| MBAR | 484 | PORTPMSC1 | Port Power Management Status and Control USB2 |
| MBAR | 48C | PORTHLPMC1 | Port X Hardware LPM Control Register |
| MBAR | 490 | PORTSC2 | Port Status and Control USB2 |
| MBAR | 494 | PORTPMSC2 | Port Power Management Status and Control USB2 |
| MBAR | 49C | PORTHLPMC2 | Port X Hardware LPM Control Register |
| MBAR | 4A0 | PORTSC3 | Port Status and Control USB2 |
| MBAR | 4A4 | PORTPMSC3 | Port Power Management Status and Control USB2 |
| MBAR | 4AC | PORTHLPMC3 | Port X Hardware LPM Control Register |
| MBAR | 4B0 | PORTSC4 | Port Status and Control USB2 |
| MBAR | 4B4 | PORTPMSC4 | Port Power Management Status and Control USB2 |
| MBAR | 4BC | PORTHLPMC4 | Port X Hardware LPM Control Register |
| MBAR | 4C0 | PORTSC5 | Port Status and Control USB3 |
| MBAR | 4C4 | PORTPMSC5 | Port Power Management Status and Control USB3 |
| MBAR | 4C8 | PORTLI5 | USB3 Port Link Info |
| MBAR | 4D0 | PORTSC6 | Port Status and Control USB3 |
| MBAR | 4D4 | PORTPMSC6 | Port Power Management Status and Control USB3 |
| MBAR | 4D8 | PORTLI6 | USB3 Port Link Info |
| MBAR | 4E0 | PORTSC7 | Port Status and Control USB3 |
| MBAR | 4E4 | PORTPMSC7 | Port Power Management Status and Control USB3 |
| MBAR | 4E8 | PORTLI7 | USB3 Port Link Info |
| MBAR | 4F0 | PORTSC8 | Port Status and Control USB3 |
| MBAR | 4F4 | PORTPMSC8 | Port Power Management Status and Control USB3 |
| MBAR | 4F8 | PORTLI8 | USB3 Port Link Info |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 3 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **Host Controller Runtime Registers** | | | |
| MBAR | 2000 | RTMFINDEX | Microframe Index |
| MBAR | 2020 | IMAN0 | Interrupter 1 Management |
| MBAR | 2024 | IMOD0 | Interrupter 1 Moderation |
| MBAR | 2028 | ERSTSZ0 | Event Ring Segment Table Size 1 |
| MBAR | 2030 | ERSTBA_LO0 | Event Ring Segment Table Base Address Low 1 |
| MBAR | 2034 | ERSTBA_HI0 | Event Ring Segment Table Base Address High 1 |
| MBAR | 2038 | ERDP_LO0 | Event Ring Dequeue Pointer Low 1 |
| MBAR | 203C | ERDP_HI0 | Event Ring Dequeue Pointer High 1 |
| MBAR | 2040 | IMAN1 | Interrupter 2 Management |
| MBAR | 2044 | IMOD1 | Interrupter 2 Moderation |
| MBAR | 2048 | ERSTSZ1 | Event Ring Segment Table Size 2 |
| MBAR | 2050 | ERSTBA_LO1 | Event Ring Segment Table Base Address Low 2 |
| MBAR | 2054 | ERSTBA_HI1 | Event Ring Segment Table Base Address High 2 |
| MBAR | 2058 | ERDP_LO1 | Event Ring Dequeue Pointer Low 2 |
| MBAR | 205C | ERDP_HI1 | Event Ring Dequeue Pointer High 2 |
| MBAR | 2060 | IMAN2 | Interrupter 3 Management |
| MBAR | 2064 | IMOD2 | Interrupter 3 Moderation |
| MBAR | 2068 | ERSTSZ2 | Event Ring Segment Table Size 3 |
| MBAR | 2070 | ERSTBA_LO2 | Event Ring Segment Table Base Address Low 3 |
| MBAR | 2074 | ERSTBA_HI2 | Event Ring Segment Table Base Address High 3 |
| MBAR | 2078 | ERDP_LO2 | Event Ring Dequeue Pointer Low 3 |
| MBAR | 207C | ERDP_HI2 | Event Ring Dequeue Pointer High 3 |
| MBAR | 2080 | IMAN3 | Interrupter 4 Management |
| MBAR | 2084 | IMOD3 | Interrupter 4 Moderation |
| MBAR | 2088 | ERSTSZ3 | Event Ring Segment Table Size 4 |
| MBAR | 2090 | ERSTBA_LO3 | Event Ring Segment Table Base Address Low 4 |
| MBAR | 2094 | ERSTBA_HI3 | Event Ring Segment Table Base Address High 4 |
| MBAR | 2098 | ERDP_LO3 | Event Ring Dequeue Pointer Low 4 |
| MBAR | 209C | ERDP_HI3 | Event Ring Dequeue Pointer High 4 |
| MBAR | 20A0 | IMAN4 | Interrupter 5 Management |
| MBAR | 20A4 | IMOD4 | Interrupter 5 Moderation |
| MBAR | 20A8 | ERSTSZ4 | Event Ring Segment Table Size 5 |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 4 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **Host Controller Runtime Registers (continued)** | | | |
| MBAR | 20B0 | ERSTBA_LO4 | Event Ring Segment Table Base Address Low 5 |
| MBAR | 20B4 | ERSTBA_HI4 | Event Ring Segment Table Base Address High 5 |
| MBAR | 20B8 | ERDP_LO4 | Event Ring Dequeue Pointer Low 5 |
| MBAR | 20BC | ERDP_HI4 | Event Ring Dequeue Pointer High 5 |
| MBAR | 20C0 | IMAN5 | Interrupter 6 Management |
| MBAR | 20C4 | IMOD5 | Interrupter 6 Moderation |
| MBAR | 20C8 | ERSTSZ5 | Event Ring Segment Table Size 6 |
| MBAR | 20D0 | ERSTBA_LO5 | Event Ring Segment Table Base Address Low 6 |
| MBAR | 20D4 | ERSTBA_HI5 | Event Ring Segment Table Base Address High 6 |
| MBAR | 20D8 | ERDP_LO5 | Event Ring Dequeue Pointer Low 6 |
| MBAR | 20DC | ERDP_HI5 | Event Ring Dequeue Pointer High 6 |
| MBAR | 20E0 | IMAN6 | Interrupter 7 Management |
| MBAR | 20E4 | IMOD6 | Interrupter 7 Moderation |
| MBAR | 20E8 | ERSTSZ6 | Event Ring Segment Table Size 7 |
| MBAR | 20F0 | ERSTBA_LO6 | Event Ring Segment Table Base Address Low 7 |
| MBAR | 20F4 | ERSTBA_HI6 | Event Ring Segment Table Base Address High 7 |
| MBAR | 20F8 | ERDP_LO6 | Event Ring Dequeue Pointer Low 7 |
| MBAR | 20FC | ERDP_HI6 | Event Ring Dequeue Pointer High 7 |
| MBAR | 2100 | IMAN7 | Interrupter 8 Management |
| MBAR | 2104 | IMOD7 | Interrupter 8 Moderation |
| MBAR | 2108 | ERSTSZ7 | Event Ring Segment Table Size 8 |
| MBAR | 2110 | ERSTBA_LO7 | Event Ring Segment Table Base Address Low 8 |
| MBAR | 2114 | ERSTBA_HI7 | Event Ring Segment Table Base Address High 8 |
| MBAR | 2118 | ERDP_LO7 | Event Ring Dequeue Pointer Low 8 |
| MBAR | 211C | ERDP_HI7 | Event Ring Dequeue Pointer High 8 |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 5 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **Doorbell Registers** | | | |
| MBAR | 3000 | DB0 | Door Bell 1 |
| MBAR | 3004 | DB1 | Door Bell 2 |
| MBAR | 3008 | DB2 | Door Bell 3 |
| MBAR | 300C | DB3 | Door Bell 4 |
| MBAR | 3010 | DB4 | Door Bell 5 |
| MBAR | 3014 | DB5 | Door Bell 6 |
| MBAR | 3018 | DB6 | Door Bell 7 |
| MBAR | 301C | DB7 | Door Bell 8 |
| MBAR | 3020 | DB8 | Door Bell 9 |
| MBAR | 3024 | DB9 | Door Bell 10 |
| MBAR | 3028 | DB10 | Door Bell 11 |
| MBAR | 302C | DB11 | Door Bell 12 |
| MBAR | 3030 | DB12 | Door Bell 13 |
| MBAR | 3034 | DB13 | Door Bell 14 |
| MBAR | 3038 | DB14 | Door Bell 15 |
| MBAR | 303C | DB15 | Door Bell 16 |
| MBAR | 3040 | DB16 | Door Bell 17 |
| MBAR | 3044 | DB17 | Door Bell 18 |
| MBAR | 3048 | DB18 | Door Bell 19 |
| MBAR | 304C | DB19 | Door Bell 20 |
| MBAR | 3050 | DB20 | Door Bell 21 |
| MBAR | 3054 | DB21 | Door Bell 22 |
| MBAR | 3058 | DB22 | Door Bell 23 |
| MBAR | 305C | DB23 | Door Bell 24 |
| MBAR | 3060 | DB24 | Door Bell 25 |
| MBAR | 3064 | DB25 | Door Bell 26 |
| MBAR | 3068 | DB26 | Door Bell 27 |
| MBAR | 306C | DB27 | Door Bell 28 |
| MBAR | 3070 | DB28 | Door Bell 29 |
| MBAR | 3074 | DB29 | Door Bell 30 |
| MBAR | 3078 | DB30 | Door Bell 31 |
| MBAR | 307C | DB31 | Door Bell 32 |
| MBAR | 3080 | DB32 | Door Bell 32 |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 6 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **xHCI Extended Capability Pointer Registers** | | | |
| MBAR | 8000 | XECP_SUPP_USB2_0 | XECP_SUPP_USB2_0 |
| MBAR | 8004 | XECP_SUPP_USB2_1 | XECP_SUPP_USB2_1 |
| MBAR | 8008 | XECP_SUPP_USB2_2 | XECP_SUPP_USB2_2 |
| MBAR | 8010 | XECP_SUPP_USB2_3 | XECP_SUPP_USB2_3 (Full Speed) |
| MBAR | 8014 | XECP_SUPP_USB2_4 | XECP_SUPP_USB2_4 (Low Speed) |
| MBAR | 8018 | XECP_SUPP_USB2_5 | XECP_SUPP_USB2_5 (High Speed) |
| MBAR | 8020 | XECP_SUPP_USB3_0 | XECP_SUPP_USB3_0 |
| MBAR | 8024 | XECP_SUPP_USB3_1 | XECP_SUPP_USB3_1 |
| MBAR | 8028 | XECP_SUPP_USB3_2 | XECP_SUPP_USB3_2 |
| MBAR | 8030 | XECP_SUPP_USB3_3 | XECP_SUPP_USB3_3 |
| MBAR | 8034 | XECP_SUPP_USB3_4 | XECP_SUPP_USB3_4 |
| MBAR | 8038 | XECP_SUPP_USB3_5 | XECP_SUPP_USB3_5 |
| MBAR | 803C | XECP_SUPP_USB3_6 | XECP_SUPP_USB3_6 |
| MBAR | 8040 | XECP_SUPP_USB3_7 | XECP_SUPP_USB3_7 |
| MBAR | 8044 | XECP_SUPP_USB3_8 | XECP_SUPP_USB3_8 |
| MBAR | 8048 | XECP_SUPP_USB3_9 | XECP_SUPP_USB3_9 |
| **xHCI Extended Capability Pointer Registers** | | | |
| MBAR | 8070 | HOST_CTRL_CAP_REG | Host Controller Capability |
| MBAR | 8078 | HOST_CLR_MASK_REG | Override EP Flow Control |
| MBAR | 807C | HOST_CLR_IN_EP_VALID_REG | Clear Active IN EP ID Control |
| MBAR | 8080 | HOST_CLR_PMASK_REG | Clear Poll Mask Control |
| MBAR | 8094 | HOST_CTRL_SCH_REG | Host Control Scheduler |
| MBAR | 80A0 | HOST_CTRL_PORT_CTRL | Global Port Control |
| MBAR | 80A4 | PMCTRL_REG | Power Management Control |
| MBAR | 80A8 | PGCBCTRL_REG | PGCB Control |
| MBAR | 80AC | DOI3CTRL_REG | D0I3 Control |
| MBAR | 80B0 | HOST_CTRL_MISC_REG | HOST_CTRL_MISC_REG |
| MBAR | 80B4 | HOST_CTRL_MISC_REG2 | HOST_CTRL_MISC_REG2 |
| MBAR | 80B8 | SSPE_REG | SSPE_REG |
| MBAR | 80BC | SSPITPE | |
| MBAR | 80C0 | AUX_CTRL_REG | AUX Reset Control |
| MBAR | 80C4 | HOST_BW_OV_SS_REG | Super Speed Bandwidth Overload |
| MBAR | 80C8 | HOST_BW_OV_HS_REG | High Speed TT Bandwidth Overload |
| MBAR | 80CC | HOST_BW_OV_FS_LS_REG | Bandwidth Overload Full Low Speed |
| MBAR | 80D0 | HOST_BW_OV_SYS_REG | System Bandwidth Overload |
| MBAR | 80D4 | HOST_CTRL_SCH_ASYNC_DELAY_REG | Scheduler Async Delay |
| MBAR | 80E0 | AUX_CTRL_REG1 | AUX Power Management Control |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 7 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **xHCI Extended Capability Pointer Registers (continued)** | | | |
| MBAR | 80E4 | BATTERY_CHARGE_REG | Battery Charge |
| MBAR | 80E8 | HOST_CTRL_WATERMARK_REG | Port Watermark |
| MBAR | 80EC | HOST_CTRL_PORT_LINK_REG | SuperSpeed Port Link Control |
| MBAR | 80F0 | USB2_LINK_MGR_CTRL_REG1 | USB2 Port Link Control 1 |
| MBAR | 80F4 | USB2_LINK_MGR_CTRL_REG2 | USB2 Port Link Control 2 |
| MBAR | 80F8 | USB2_LINK_MGR_CTRL_REG3 | USB2 Port Link Control 3 |
| MBAR | 80FC | USB2_LINK_MGR_CTRL_REG4 | USB2 Port Link Control 4 |
| MBAR | 8100 | HOST_CTRL_BW_CTRL_REG | Bandwidth Calc Control |
| MBAR | 8108 | HOST_IF_CTRL_REG | Host Interface Control |
| MBAR | 810C | HOST_BW_OV_BURST_REG | Bandwidth Overload Burst |
| MBAR | 8128 | HOST_CTRL_BW_MAX_REG | USB Max Bandwidth Control 4 |
| MBAR | 8130 | LINESTATE_DEBUG_REG | USB2 Linestate Debug |
| MBAR | 8134 | USB2_PROTOCOL_GAP_TIMER_REG | USB2 Protocol Gap Timer |
| MBAR | 813C | USB2_PROTOCOL_BTO_TIMER_REG | USB2 Protocol Bus Timeout Timer |
| MBAR | 8140 | PWR_SCHED_CTRL0 | Power Scheduler Control-0 |
| MBAR | 8144 | PWR_SCHED_CTRL2 | Power Scheduler Control-2 |
| MBAR | 8154 | AUX_CTRL_REG2 | AUX Power Management Control |
| MBAR | 8164 | USB2_PHY_PMC | USB2 PHY Power Management Control |
| MBAR | 8168 | USB_PGC | USB Power Gating Control |
| MBAR | 816C | XHCI_AUX_CCR | xHCI Aux Clock Control Register |
| MBAR | 8170 | USB_LPM_PARAM | USB LPM Parameters |
| MBAR | 8174 | XLTP_LTV1 | xHC Latency Tolerance Parameters - LTV Control |
| MBAR | 8178 | XLTP_LTV2 | xHC Latency Tolerance Parameters LTV Control 2 |
| MBAR | 817C | XLTP_HITC | xHC Latency Tolerance Parameters - High Idle Time Control |
| MBAR | 8180 | XLTP_MITC | xHC Latency Tolerance Parameters - Medium Idle Time Control |
| MBAR | 8184 | XLTP_LITC | xHC Latency Tolerance Parameters Low Idle Time Control |
| MBAR | 8188 | HOST_CTRL_BW_MAX3_REG | HOST_CTRL_BW_MAX3_REG |
| MBAR | 8198 | PDDIS_REG | PDDIS_REG |
| MBAR | 819C | THRM_HOST_CTRL_REG | THRM_HOST_CTRL_REG |
| MBAR | 81A0 | LFPS_PM_CTRL_REG | LFPS_PM_CTRL_REG |
| MBAR | 81A4 | U2PDM | U2PDM |
| MBAR | 81A8 | U2PCM | U2PCM |
| MBAR | 81AC | U3PDM | U3PDM |
| MBAR | 81B0 | U3PCM | U3PCM |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 8 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **xHCI Extended Capability Pointer Registers (continued)** | | | |
| MBAR | 81B4 | THRM_HOST_CTRL_REG2 | THRM_HOST_CTRL_REG2 |
| MBAR | 81B8 | LFPSONCOUNT_REG | LFPSONCOUNT_REG |
| MBAR | 81BC | D0i2_CTRL_REG | |
| MBAR | 81C0 | D0i2_SCH_ALARM_CTRL_REG | |
| MBAR | 81C4 | USB2PMCTRL_REG | |
| MBAR | 83F8 | ECC_PARITY_ERROR_LOG_REG | ECC_PARITY_ERROR_LOG_REG |
| MBAR | 83FC | ECC_POISONING_CTRL_REG | ECC_POISONING_CTRL_REG |
| MBAR | 8400 | USB2_PORT_STATE_REG | USB2_PORT_STATE_REG |
| MBAR | 8408 | USB3_PORT_STATE_REG | USB2_PORT_STATE_REG |
| MBAR | 8410 | FUS1_REG | FUS1_REG |
| MBAR | 8414 | FUS2_REG | FUS2_REG |
| MBAR | 8418 | FUS3_REG | FUS3_REG |
| MBAR | 841C | STRAP1_REG | STRAP1_REG |
| MBAR | 8420 | STRAP2_REG | STRAP2_REG |
| MBAR | 8424 | STRAP3_REG | STRAP3_REG |
| MBAR | 8430 | DFT_REG1 | DFT_REG1 |
| MBAR | 8434 | DFT_REG2 | DFT_REG2 |
| MBAR | 8438 | DFT_REG3 | DFT_REG3 |
| MBAR | 843C | DFT_REG4 | dft_reg4 |
| MBAR | 8440 | DFT_REG5 | dft_reg5 |
| MBAR | 8448 | XECP_CMDM_STS0 | XECP_CMDM_STS0 |
| MBAR | 844C | XECP_CMDM_STS1 | XECP_CMDM_STS1 |
| MBAR | 8450 | XECP_CMDM_STS2 | XECP_CMDM_STS2 |
| MBAR | 8454 | XECP_CMDM_STS3 | XECP_CMDM_STS3 |
| MBAR | 8458 | XECP_CMDM_STS4 | XECP_CMDM_STS4 |
| MBAR | 845C | XECP_CMDM_STS5 | XECP_CMDM_STS5 |
| MBAR | 8460 | UPORTS_PON_RST_REG | AUX Power PHY Reset |
| MBAR | 8464 | HOST_IF_LAT_TOL_CTRL_REG0 | Latency Tolerance Control 0 |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 9 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **USB Legacy Support Capability Registers** | | | |
| MBAR | 846C | USBLEGSUP | USB Legacy Support Capability |
| MBAR | 8470 | USBLEGCTLSTS | USB Legacy Support Control Status |
| **USB Port Disable Override Capability Registers** | | | |
| MBAR | 84F4 | PDO_CAPABILITY | Port Disable Override capability register |
| MBAR | 84F8 | USB2PDO | USB2 Port Disable Override |
| MBAR | 84FC | USB3PDO | USB3 Port Disable Override |
| **Hardware State Capability Registers** | | | |
| MBAR | 8500 | HW_STATE_CAPABILITY | HW state capability register |
| MBAR | 8504 | HW_STATE_REG1 | HW state register 1 |
| MBAR | 8508 | HW_STATE_REG2 | HW state register 2 |
| MBAR | 850C | HW_STATE_REG3 | HW state register 3 |
| MBAR | 8510 | HW_STATE_REG4 | HW state register 4 |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 10 of 11)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **Configuration Mirror Capability Registers** | | | |
| MBAR | 8600 | CONFIG_MIRROR_CAPABILITY | CONFIG mirror capability register |
| MBAR | 8604 | CMD_MMIO | Command |
| MBAR | 8606 | STS_MMIO | Device Status |
| MBAR | 8608 | RID_MMIO | Revision ID |
| MBAR | 8609 | PI_MMIO | Programming Interface |
| MBAR | 860A | SCC_MMIO | Sub Class Code |
| MBAR | 860B | BCC_MMIO | Base Class Code |
| MBAR | 860D | MLT_MMIO | Master Latency Timer |
| MBAR | 860E | HT_MMIO | Header Type |
| MBAR | 8610 | MBAR_MMIO | Memory Base Address |
| MBAR | 862E | SSID_MMIO | USB Subsystem ID |
| MBAR | 8634 | CAP_PTR_MMIO | Capabilities Pointer |
| MBAR | 863C | ILINE_MMIO | Interrupt Line |
| MBAR | 863D | IPIN_MMIO | Interrupt Pin |
| MBAR | 8640 | XHCC1_MMIO | XHC System Bus Configuration 1 |
| MBAR | 8644 | XHCC2_MMIO | XHC System Bus Configuration 2 |
| MBAR | 8650 | XHCLKGTEN_MMIO | Clock Gating |
| MBAR | 8658 | AUDSYNC_MMIO | Audio Time Synchronization |
| MBAR | 8660 | SBRN_MMIO | Serial Bus Release Number |
| MBAR | 8661 | FLADJ_MMIO | Frame Length Adjustment |
| MBAR | 8662 | BESL_MMIO | Best Effort Service Latency |
| MBAR | 8670 | PM_CID_MMIO | PCI Power Management Capability ID |
| MBAR | 8671 | PM_NEXT_MMIO | Next Item Pointer #1 |
| MBAR | 8672 | PM_CAP_MMIO | Power Management Capabilities |
| MBAR | 8674 | PM_CS_MMIO | Power Management Control/Status |
| MBAR | 8680 | MSI_CID_MMIO | Message Signaled Interrupt CID |
| MBAR | 8681 | MSI_NEXT_MMIO | Next item pointer |
| MBAR | 8682 | MSI_MCTL_MMIO | Message Signaled Interrupt Message Control |
| MBAR | 8684 | MSI_MAD_MMIO | Message Signaled Interrupt Message Address |
| MBAR | 8688 | MSI_MUAD_MMIO | Message Signaled Interrupt Upper Address |
| MBAR | 868C | MSI_MD_MMIO | Message Signaled Interrupt Message Data |
| MBAR | 8690 | DEVIDLE_MMIO | Device Idle Capability |
| MBAR | 8694 | VSHDR_MMIO | Vendor Specific Header |
| MBAR | 8698 | SWLTRPTR_MMIO | SW LTR POINTER |
| MBAR | 869C | DEVIDLEPTR_MMIO | Device Idle Pointer Register |

## Table 14-8. Relocatable Memory-Mapped Registers (Sheet 11 of 11)

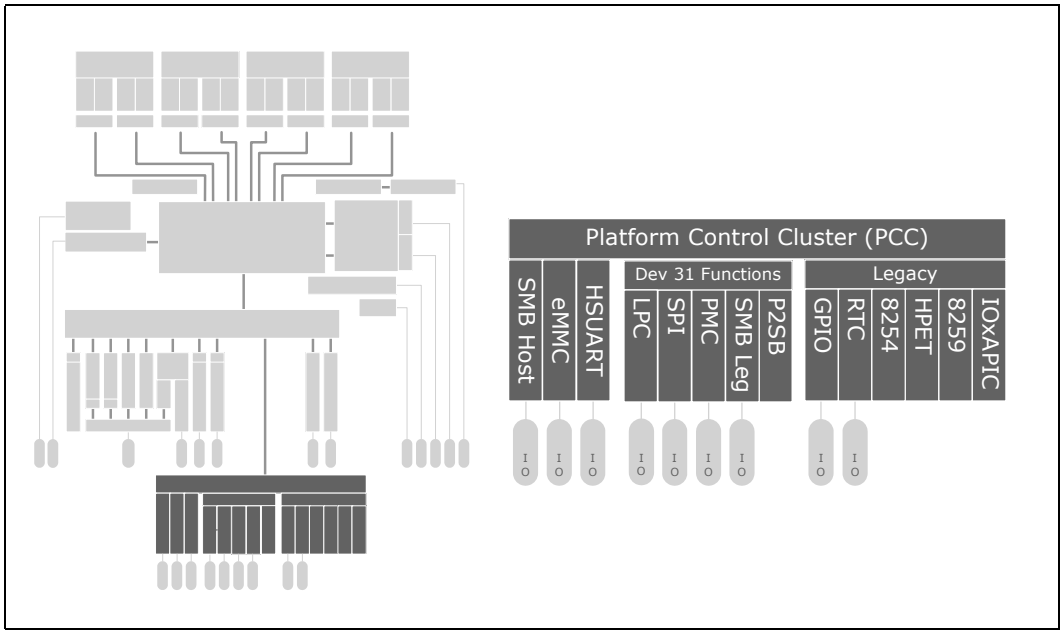
| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **Configuration Mirror Capability Registers (continued)** | | | |
| MBAR | 86A0 | DEVIDLEPOL_MMIO | Device Idle Power ON Latency |
| MBAR | 86A4 | HSCFG2_MMIO | High Speed Configuration 2 |
| MBAR | 86B0 | U2OCM1_MMIO | XHCI USB2 Overcurrent Pin Mapping 1 |
| MBAR | 86D0 | U3OCM1_MMIO | XHCI USB3 Overcurrent Pin Mapping 1 |
| MBAR | 86FC | XHCC3_MMIO | XHCC3 |
| **Debug Capability Registers** | | | |
| MBAR | 8700 | DCID | Debug Capability ID Register |
| MBAR | 8704 | DCDB | Debug Capability Doorbell Register |
| MBAR | 8708 | DCERSTSZ | Debug Capability Event Ring Segment Table Size Register |
| MBAR | 8710 | DCERSTBA | Debug Capability Event Ring Segment Table Base Address Register |
| MBAR | 8718 | DCERDP | Debug Capability Event Ring Dequeue Pointer Register |
| MBAR | 8720 | DCCTRL | Debug Capability Control Register |
| MBAR | 8724 | DCST | Debug Capability Status Register |
| MBAR | 8728 | DCPORTSC | Debug Capability Port Status and Control Register |
| MBAR | 8730 | DCCP | Debug Capability Context Pointer Register |
| MBAR | 8738 | DCDDI1 | Debug Capability Device Descriptor Info Register 1 |
| MBAR | 873C | DCDDI2 | Debug Capability Device Descriptor Info Register 2 |
| MBAR | 8740 | DCDP | Debug Capability Descriptor Parameters |
| MBAR | 8748 | DBGDEV_CTRL_ODMA_REG | Debug Device Control ODMA |
| MBAR | 8760 | DBCCTL_REG | DBC Control Register 1 |

§ §

# 15 Platform Control Cluster (PCC)

This chapter describes the Platform Control Cluster (PCC).

**Figure 15-1. What is Covered in This Chapter**



## 15.1 Signal Descriptions

The signal pins and signal names associated with this chapter are described in the chapter of each PCC device.

## 15.2 Architectural Overview

The Platform Control Cluster (PCC) consists of the devices and controllers shown in Table 15-1. Details are in the Chapters shown in the table.

**Table 15-1. Platform Control Cluster (PCC)**

| Device/Controller | Chapter |
|---|---|
| **Integrated PCI devices** | |
| SMBus - Host | See Chapter 29, "SMBus Controller - Host" for details. |
| Embedded Multimedia Card (eMMC) | See Chapter 17, "Embedded Multi-Media Card (eMMC)" for details. |
| High-Speed UART (HSUART) | See Chapter 18, "High-Speed UART Controller" for details. |
| **Device 31 Functions** | |
| LPC | See Chapter 20, "Low Pin Count (LPC) Controller" for details. |
| Serial Peripheral Interface (SPI) | See Chapter 19, "Serial Peripheral Interface (SPI)" for details. |
| Power Management Controller (PMC) | See Chapter 16, "Power Management Controller (PMC)" for details. |
| SMBus - Legacy | See Chapter 28, "SMBus Controller - Legacy" for details. |
| Primary to Sideband Bridge (P2SB) | See Chapter 22, "Primary to Side Band (P2SB) Bridge" for details. |
| **Legacy controllers** | |
| Customer GPIO | See Chapter 26, "Customer General-Purpose I/O (GPIO)" for details. |
| Real Time Clock (RTC) | See Chapter 24, "Real Time Clock (RTC)" for details. |
| 8245 Programmable Interrupt Timer (PIT) | See Chapter 23, "Programmable Interval Timer (PIT)" for details. |
| High Performance Event Timer (HPET) | See Chapter 25, "High Precision Event Timer (HPET)" for details. |
| 8259 Programmable Interrupt Controller (PIC) | See Chapter 21, "PIC and I/O APIC Controllers" for details. |
| IOxAPIC | See Chapter 21, "PIC and I/O APIC Controllers" for details. |

### 15.2.1 Integrated PCI Devices

The PCC contains three integrated PCI devices which are accessible in Host Root Space:

- Device 18 (decimal), Function 0 - SMBus (Host) controller
- Device 26 (decimal), Functions 0, 1, and 2 - High-Speed UART (HSUART) controller
- Device 28 (decimal), Function 0 - Embedded Multimedia Card (eMMC) controller

Table 15-1 on page 523 provides a list of chapters for each controller.

### 15.2.2 Integrated PCI Device 31 Functions

Accessible in Host Root Space, the integrated, PCI Device 31 (decimal), has five PCI Functions:

- Function 0:
  - Low Pin Count (LPC) controller if the "Reserved for Intel" tied to pin (MCERR_N) SoC Hard Strap pin is 0.
- Function 1 - Primary to Sideband Bridge (P2SB) controller.
- Function 2 - Power Management Controller (PMC).
- Function 3 - SMBus (Legacy) controller.
- Function 4 - Serial Peripheral Interface (SPI) controller.

Table 15-1 on page 523 provides a list of chapters for each controller.

### 15.2.3 Legacy Controller Group

The Legacy group of controllers are the Customer General-Purpose I/O (GPIO) controller, Real-Time Clock, and the group of controllers that comprise the Interrupt Timer Sub System (ITSS). Table 15-1 on page 523 provides a list of chapters for each controller. The following section summarizes the ITSS group.

#### 15.2.3.1 Interrupt Timer Sub System (ITSS)

The ITSS is a portion of the SoC internal circuitry consisting of the HPET, 8254 PIT, 8259 PIC, IOxAPIC and CPU interface components. It also provides the SoC interrupt routing control for the Host Root Space. The ITSS has its own clock and reset-control circuitry.

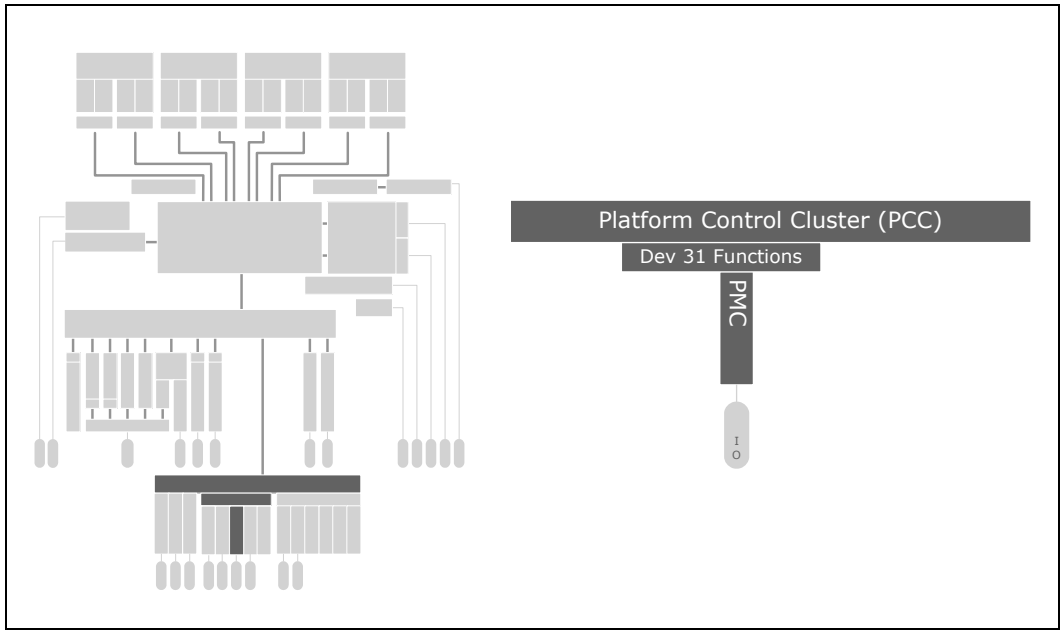
The Innovation Engine (IE) has a sideband connection to the ITSS components.

§ §

# 16 Power Management Controller (PMC)

This chapter describes the Power Management Controller (PMC) of the SoC.

**Figure 16-1. What is Covered in This Chapter**



## 16.1 Signal Descriptions

The signal descriptions are shown in Table 16-1, "Signal Names and Descriptions". For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing". The Direction/Type column of Table 16-1, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 16-1. Signal Names and Descriptions (Sheet 1 of 2)**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| ADR_TRIGGER | I,O-OD | Yes | **Asynchronous DRAM Refresh (ADR) Trigger:** This active-high input is driven by the platform board circuitry to start the SoC ADR flow of events. As an active-high, open-drain output, the SoC can also drive this signal high to trigger ADR when an enabled, internal ADR event occurs. The SoC drives this signal pin high as long as the internal, ADR-triggered event is valid.<br>The platform board must connect a pull-down resistor to this pin.<br>***Note:*** The platform board must refrain from actively driving this signal before power loss (external qualified event) has happened. The internal SoC circuitry also drives this pin active when it detects and handles an internal qualified event. |
| ADR_COMPLETE | I,O | Yes | **Asynchronous DRAM Refresh (ADR) Complete:** This active-high output indicates that the SoC has completed the ADR flow of events. When power is invalid or removed from the SoC, this output signal is invalid.<br>***Note:*** Even though this signal pin is labeled as an input-output (I/O), the platform must use it as an output-only pin and must not drive it. |
| CPU_RESET_N | O | Yes | **CPU Reset:** This active-low output can be used by the platform board or debug tools to indicate that the SoC is in a reset state. |
| PMU_PLTRST_N | O | Yes | **Platform Reset:** This active-low signal is asynchronously driven by the SoC and can be used as the platform board's main reset. This signal can be used by the PCI Express interface as the Fundamental Reset signal (PERST#) as defined in the PCIe base specification. |
| PMU_PWRBTN_N | I | Yes | **Power Button:** This active-low signal is debounced (16 ms) by the SoC. It generates an SMI or SCI to request the system to go to a sleep state. If already in sleep state, it causes a Wake Event.<br>When the signal is held asserted for 5.1 seconds or longer while the SoC is in the G0 or G1 state, a power-button override is generated. This causes an unconditional transition to the G2 soft-off state and is useful when the system is in a hung working state. |
| PMU_RESETBUTTON_N | I | Yes | **Reset Button:** This edge-triggered, active-low, input signal indicates that the system reset button is depressed.<br>The SoC debounces this input signal and does nothing unless the signal is asserted for at least 16 ms, at which time the PMC begins reset-sequence actions. When this edge-triggered signal is asserted beyond the 16-ms period, no additional reset actions are taken (only one reset produced per asserted edge).<br>***Note:*** There is a SoC Soft Strap that can be set to either enable this 16-ms debounce period or disable the internal debounce circuitry on this pin. See Chapter 4, "Strapping and Configuration". |

## Table 16-1. Signal Names and Descriptions (Sheet 2 of 2)

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| PMU_SLP_S3_N | O | Yes | **Sleep State 3:** This active-low signal indicates that the SoC is transitioning from S0 to S4 or S5, or transitioning back up to S0, and that the platform board can shut down power not needed in this state by the SoC and the rest of the system. |
| PMU_SLP_S45_N | O | Yes | **Sleep State 4 or 5:** This active-low signal indicates that the SoC has entered and is in either the Suspend-to-Disk (S4) or Soft-Off (S5) Sleep State and that the platform board can shut down power not needed in these states by the SoC and the rest of the system. |
| PMU_SUSCLK | O | Yes | **Suspend Mode Clock:** Output of the RTC circuit (32.768 kHz) that can be used by the platform board. The output has a duty cycle as low as 30% or as high as 70%. |
| PMU_WAKE_N | I | Yes | **Wake:** This active-low signal is the destination of the PCIe sideband WAKE# signals. All device WAKE# signals are open-drain output signals and are connected together as described in the PCIe base specification. |
| SUS_STAT_N | O | Yes | **Suspend Status:** This active-low signal indicates that the SoC will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off state. |
| SUSPWRDNACK | O | Yes | **Suspend Power-Down Acknowledge:** This active-high signal is asserted by the SoC when it does not require its Suspend (SUS) power well to be powered. |
| COREPWROK | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| INTRUDER_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| RSMRST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| RTEST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| SRTCRST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| THERMTRIP_N | O-OD | Yes | See Chapter 35, "Thermal Management" for details. |

## 16.2 Feature List

- ACPI and general Power Management control
- Circuitry to lower the Total Cost of Ownership (TCO) of the system
- Interacts with the SoC Innovation Engine (IE)
- Interacts with the SoC Intel® Management Engine (Intel® ME)
- Asynchronous DRAM Refresh (ADR)

## 16.3 Architectural Overview

The SoC Power Management Controller (PMC) is circuitry that provides the power management interface to the SoC signal pins. These are listed in Table 16-1, "Signal Names and Descriptions". The controller is an internal micro-controller which is not directly accessible to the SoC customer.

The PMC appears as a PCI Bus 0, Device 31 (decimal), Function 2 to software.

A number of SoC power-control and status registers are located in relocatable Host Root I/O Space starting at the ACPI Base Address (ABASE). These I/O Space registers must be enabled by setting the ACPI Enable (EN), bit 7 of the ACPI Control (ACTL) register at offset 44h of the PMC Configuration Space.

There are also a number of PMC registers located in relocatable Host Root Memory Space starting at the PM Base Address (PWRMBASE). These Memory-Mapped registers must be enabled by setting the PWRM Enable (PWRM_EN), bit 8 of the ACPI Control (ACTL) register at offset 44h of the PMC Configuration Space.

Bits [2:0] of the aforementioned ACPI Control (ACTL) register specify to which IRQ the System Control Interrupt (SCI) is routed internal to the SoC. The default is IRQ9 and other options are programmable.

The PMC performs the software-generated reset when the Host software writes a value of 06h or 0Eh to the Reset Control (RST_CNT) Register located at address CF9h in I/O Space.

The Total Cost of Ownership (TCO) registers are accessible and are located in the SMBus-Legacy Controller that appears as a PCI Bus 0, Device 31 (decimal), Function 4 to software.

## 16.4 System Power States

See Section 34.8, "System Global Power States" on page 911.

## 16.5 SoC Internal Power Control

The required supply voltages for the SoC operation for each Global Power State are shown in Chapter 34, "Supply Voltage Rails". The PMC platform board must provide the SoC voltages, but the PMC circuitry controls the internal silicon voltage rails for its Power Wells, turning internal circuitry on and off as needed. Whether a platform board turns its voltage regulators on and off is up to the system designer, but the regulators must provide power during the power states shown in Chapter 34, "Supply Voltage Rails".

### 16.5.1 Internal Power Switching

The SoC internal circuitry required for a particular product SKU is fully powered-on during the ACPI G0 (S0) state. The internal SoC circuitry that is only powered-on during the G0 (S0) working state is in the Core Power Well. See Section 34.4.1, "Core Power Well" on page 906. Some internal functions of a product SKU can be programmed to be powered off.

When the system is in the G2 (S5) Soft Off state, the PMC turns off power to some of the internal circuitry. The internal circuitry that is powered-on during G0 (S0) and during the G2 (S5) state is in the SUS Power Well. See Section 34.4.2, "SUS Power Well" on page 906. Sometimes this is referred to as "Always On" circuitry. The integrated circuitry that is powered-on during G2 (S5) is shown Table 16-2, "SoC Circuitry Powered-On During the G2 (S5) Global Power State".

**Table 16-2. SoC Circuitry Powered-On During the G2 (S5) Global Power State**

| SoC Internal Module | Circuitry Powered On | Power Usage |
|---|---|---|
| Power Management | All | Required for SoC state changes |
| USB Combo Controller | USB 2.0 PHY and portion of the USB controller | Required for debug interface and Wakeup |
| High-Speed IO (HSIO) | All (PCI Express, SATA, USB 3.0) | Required for Wakeup |
| LAN | PHY | Required for Wakeup (if enabled) |
| Customer General Purpose I/O (GPIO) | All | Required for Wakeup and operation in the G2 (S5) state |
| Intel® Management Engine | All | |
| Innovation Engine | All | |
| Intel® Trace Hub | All | |

When all power supplied to the SoC is removed, the system is in the G3 Mechanical Off state. If the system design includes a functioning RTC battery, a number of SoC registers draw power from the battery and retain their information during the Mechanical-Off state. These registers are also powered-on during the G0 (S0) and G2 (S5) states and are in the SoC RTC Power Well. See Section 34.4.3, "RTC Power Well" on page 906. The integrated circuitry that is powered-on during G3 is shown Table 16-3, "SoC Circuitry Powered-On During the G3 Global Power State".

**Table 16-3. SoC Circuitry Powered-On During the G3 Global Power State**

| SoC Internal Module | Circuitry Powered On | Power Usage |
|---|---|---|
| SoC RTC Circuitry | RTC Registers, 32.768 kHz clock, SRAM, time and calendar | Retain register and time information during G3 |
| Power Management | Registers preserved by a functioning RTC battery | Saving system information when system power is off |
| Intel® Management Engine | Registers preserved by a functioning RTC battery | Saving system information when system power is off |

## 16.6 Total Cost of Ownership (TCO)

The TCO system management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that some of the system management functionality can be provided without the aid of an external micro-controller or a Baseboard Management Controller (BMC).

While the TCO circuitry and registers are located in the SoC SMBus-Legacy controller, the SoC does not provide SMBus connectivity to the TCO mechanism. The SoC does allow Operating System (OS) software to access the TCO registers. TCO provides various functions to make a system easier to manage and to lower the total cost of ownership of the system. TCO features supported by the SoC:

- First TCO Watchdog Timer timeout can generate SMI# after a programmable time
  - The programmable 10-bit timer is decremented approximately every 0.6 seconds and allows timeouts ranging from 1.2 seconds to 613.8 seconds. The timeout value is programmable in the TCO_TMR Register (TTMR).
  - OS software can periodically reload the SoC timer using the TCO_RLD Register (TRLD). The first TCO Watchdog Timer timeout causes an SMI# allowing SMM-based recovery from OS lockup.
- Second hard-coded TCO Watchdog Timer timeout to generate a system reboot by the SoC asserting the active low PMU_PLTRST_N signal
  - This second timer is used only after the first TCO Watchdog Timer timeout occurs.
  - The SMI# handler must reload the first timer within 2.4 seconds after it times out to prevent the second timer from causing a system reboot. A timeout here assumes to be from a CPU or hardware error and reason to generate a system reset and reboot.
  - Option to prevent the system from rebooting after the second timeout via the "No TCO Reboot Enable" Hard Pin Strap. See Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions".
- Capability for the SMM handler to generate a TCO interrupt to the OS
- Capability for OS to generate an SMI# to alert the TCO code in the SMM handler
- Processor Present detection
  - Detects if the processor fails to fetch the first instruction after reset.
- Report various errors detected by the circuitry
  - Can generate an SMI# or TCO interrupt (SERR, SMI, SCI).
- Intruder-Detect input (INTRUDER_N):
  - Can generate a TCO interrupt or SMI# when the system cover is removed.
  - The IRQ interrupt routed per register
  - The INTRUDER_N input signal is allowed to go active in any power state, including G3.
- Illegal BIOS accesses through SPI Flash is tracked and enabled in local registers in SPI Flash.
- Detection of bad BIOS Flash programming:
  - Detects if data on first read is FFh (an indication that the BIOS flash device is not programmed).

### 16.6.1 Detecting a System Lockup

When the processor is reset, it is expected to fetch its first instruction. If the processor fails to fetch the first instruction after reset, the TCO Watchdog Timer times out twice and the SoC asserts the Platform Reset (PMU_PLTRST_N) SoC output signal.

The Platform Reset and subsequent re-boot does not occur if the "No TCO Reboot" SoC Hard Pin Strap is active when sampled when the Resume Reset RSMRST_N signal deasserts. See Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions".

During every boot, BIOS must read the SECOND_TO_STS bit in the TCO2_STS Register (TSTS2) to see if this is normal boot or a reboot due to the timeout. This bit is only cleared by writing a 1 to this bit or by assertion of the active low RSMRST_N SoC input signal pin by the platform board. The BIOS may also check the OS_POLICY bits of the TCO2_CNT Register (TCTL2) to see if it should try another boot or shutdown.

### 16.6.2 Handling an Intruder

The SoC has an active-low input signal, INTRUDER_N, that can be wired to a switch that is activated by the system's case being open. The activation of this signal can be used to cause an SMI# and to send an interrupt to the Intel® Management Engine (Intel® ME). INTRUDER_N may go active in any power state.

This SoC input has a two RTC-clock de-bounce delay. If the INTRUDER_N signal is driven active (low), and after the de-bounce delay, the Soc sets the Intruder Detect (INTRD_DET) bit of the TCO2_STS Register (TSTS2). The INTRD_SEL bits in the TCO2_CNT Register (TCTL2) can be programmed to enable the SoC to cause:

- an interrupt as selected by the TCO IRQ Select (IS) bits of the TCO Configuration (TCOCFG) register, or
- an SMI#, or
- neither (default).

The BIOS or interrupt handler can then cause a transition to the S5 state by writing to the SLP_TYP and SLP_EN bit fields of the Power Management 1 Control (PM1_CNT) register.

If the INTRUDER_N signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET bit will go to a 0 when INTRUDER_N input signal goes inactive.

The INTRUDER_N signal pin is not shared nor can it be assigned as a Customer GPIO. Even so, software can directly read the status of the INTRUDER_N signal (high or low) by clearing and then reading the INTRD_DET bit. This allows the INTRUDER_N signal pin to be used as a SoC general-purpose input (GPI) if the intruder function is not required.

*Note:* Refer to the register description shown for INTRD_DET for additional details about how software interfaces with this bit and the 65-µs recovery time involved. The bit is slightly different than a classic sticky bit, since most sticky bits remain active indefinitely when the signal goes active and immediately go inactive when a 1 is written to the bit.

### 16.6.3 SMM-to-OS and OS-to-SMM Calls

Two eight-bit data registers are provided by the SoC for interaction between an SMI handler and OS-related code.

1. The SMI handler can generate an interrupt to the OS by writing to the TCO_DAT_OUT register. Doing this sets the TCO_INT_STS bit in the TCO1_STS (TSTS1) register. It also causes an interrupt as selected by the TCO IRQ Select (IS) bits of the TCO Configuration (TCOCFG) register. The interrupt is cleared by writing a 1 to the TCO_INT_STS bit.
2. The OS (or driver) can generate an SMI# by writing to the TCO_DAT_IN Register. Doing this sets the OS_TCO_SMI bit in the TCO1_STS Register (TSTS1). The SMI# is cleared by writing a 1 to the OS_TCO_SMI bit.

Reads to the TCO_DAT_IN (TDI) and TCO_DAT_OUT (TDO) registers do not cause an SMI# or interrupt.

Writing a 1 to the NMI_NOW bit of the TCO1_CNT (TCTL1) register allows for an immediate NMI.

## 16.7 SMI#/SCI Generation

The SoC does not have an System Management Interrupt (SMI)# signal pin. Upon any enabled SMI event taking place while the End of SMI (EOS) bit of the SMI Control and Enable (SMI_EN) is set, the SoC clears the EOS bit and asserts an SMI internal signal to the processor. This causes the CPU to enter System Management Mode (SMM). The PMC takes no further action on behalf of active SMI events until the Host software sets the EOS bit again. At that point, if any SMI events are still active, the PMC sends another SMI internal signal to the processor.

The SoC must have SMI fully enabled when the SoC is enabled to trap cycles. If SMI is not enabled in conjunction with the trap enabling, then hardware behavior is undefined.

The System Control Interrupt (SCI) is a level-mode interrupt that is typically handled by an ACPI-aware operating system. The SCI IRQ Select (SCIS) bits of the ACPI Control (ACTL) register specify the IRQ to which the SCI is mapped.

In non-APIC systems (which is the default), the SCIS bits must be set to route the SCI IRQ to one of the 8259 interrupts IRQ9, IRQ10, or IRQ11. The integrated 8259 interrupt controller must be programmed to level mode for that interrupt. Here the selected IRQ cannot be shared with the SoC LPC Serialized Interrupt (LPC_SERIRQ pin) stream, but it can be shared with other PCI interrupts.

In systems using the APIC, the SCIS bits can routed the SCI to the 8259 interrupts IRQ9, IRQ10, IRQ11, and if APIC Mode is enabled, IRQ20, IRQ21, IRQ22, or IRQ23. The interrupt polarity changes depending on whether it is on an interrupt shareable with a PIRQ or not. The interrupt remains asserted until all SCI sources are removed.

For additional information see Section 8.4, “System Control Interrupt (SCI)” on page 339. Table 16-4, “Causes of SMI and SCI” shows which events can cause an SMI and SCI. The SCI_EN bit of the Power Management 1 Control (PM1_CNT) register and the GBL_SMI_EN bit of the GPE0_EN_127_96 register determine whether the interrupt is routed as an SCI (SCI_EN=1) or SMI (SCI_EN = 0 AND GBL_SMI_EN = 1).

## Table 16-4. Causes of SMI and SCI (Sheet 1 of 2)

| Cause | SCI | SMI | Additional Enables (Note 1) | Where Reported |
|---|---|---|---|---|
| PME_B0<br>*Internal PME-Capable Agents* | Yes | Yes | PME_B0_EN=1 | PME_B0_STS |
| PCI Express* PME Messages<br>*Each Root Port has two SCI enable bits in their SCICSR register in Host Configuration Space* | Yes | No | PCI_EXP_EN=1 | PCI_EXP_STS |
| PCI Express* PME Messages<br>*Each Root Port has two SMI enable bits in its SMICSR register in Host Configuration Space* | No | Yes | None | PCI_EXP_SMI_STS |
| Power Button Press | Yes | Yes | PWRBTN_EN=1 | PWRBTN_STS |
| Power Button Override (Note 2) | Yes | No | None | PRBTNOR_STS |
| RTC Alarm | Yes | Yes | RTC_EN=1 | RTC_STS |
| ACPI Timer overflow (2.34 seconds) | Yes | Yes | TMROF_EN=1 | TMROF_STS |
| Any GPIO enabled for SCI/SMI (Note 3) | Yes | Yes | | |
| TCO SCI Logic | Yes | No | TCOSCI_EN=1 | TCOSCI_STS |
| TCO SMI Logic | No | Yes | TCO_EN=1 | TCO_STS |
| TCO SMI – TCO TIMEROUT | No | Yes | None | TIMEOUT |
| TCO SMI – NMI occurred<br>*and NMIs mapped to SMI* | No | Yes | NMI2SMI_EN=1 | NMI2SMI_STS |
| TCO SMI – Write attempted to BIOS | No | Yes | BIOSWE=1 | BIOSWR_STS |
| BIOS_RLS bit of the SMI_EN register is written to 1 by BIOS (Note 4) | Yes | No | GBL_EN=1 | GBL_STS |
| GBL_RLS written to | No | Yes | BIOS_EN=1 | BIOS_STS |
| Write to B2h register | No | Yes | APMC_EN = 1 | APM_STS |
| Periodic timer expires | No | Yes | PERIODIC_EN=1 | PERIODIC_STS |
| 64 ms timer expires | No | Yes | SWSMI_TMR_EN=1 | SWSMI_TMR_STS |
| Monitor Status | No | Yes | None | MONITOR_STS |
| Enhanced USB Legacy Support Event | No | Yes | LEGACY_USB2_EN = 1 | LEGACY_USB2_STS |
| PMC instructed to generate an SCI | No | Yes | MCSMI_EN | MCSMI_STS |
| LPC SMI is trapped if trapping is enabled<br>*for legacy USB keyboard emulation* | No | Yes | LEGACY_USB_EN | LEGACY_USB_SMI_EN |
| Serialized IRQ SMI reported | No | Yes | None | SERIRQ_SMI_STS |
| SPI Command Completed | No | Yes | None | SPI_STS |
| SPI Flash Controller asserted SMI | No | Yes | None | SPI_SMI_STS |
| Device monitors match address in its range | No | Yes | None | DEVTRAP_STS |
| SMBus Host Controller | No | Yes | SMB_SMI_EN<br>*Host Controller Enabled* | SMBus Host Status Register |
| SMBus Slave SMI message | No | Yes | None | SMBUS_SMI_STS |
| SMBus SMBALERT# signal active | No | Yes | None | SMBUS_SMI_STS |
| SMBus Host Notify message received | No | Yes | HOST_NOTIFY_INTREN | SMBUS_SMI_STS<br>HOST_NOTIFY_STS |
| SLP_EN bit written to 1 | No | Yes | SMI_ON_SLP | SMI_ON_SLP_EN_STS |
| Software Generated GPE | Yes | Yes | SWGPE_EN=1 | SWGPE_STS |
| Intel® Management Engine causes SCI | Yes | No | ME_SCI_EN=1 | ME_SCI_STS |

**Table 16-4. Causes of SMI and SCI (Sheet 2 of 2)**

| Cause | SCI | SMI | Additional Enables (Note 1) | Where Reported |
|---|---|---|---|---|
| Intel® Management Engine causes SMI | No | Yes | ME_SMI_EN=1 | ME_SMI_STS |
| Innovation Engine causes SCI | Yes | No | IE_SCI_EN=1 | IE_SCI_STS |
| Innovation Engine causes SMI | No | Yes | IE_SMI_EN=1 | IE_SMI_STS |
| RTC update-in-progress | No | Yes | If enabled in the I/O Trap Register Space | RTC_UIP_SMI_STS |
| USB3 (xHCI) SMI Event | No | Yes | XHCI_SMI_EN=1 | XHCI_SMI_STS |
| Wake Alarm Device Timer | Yes | Yes | WADT_EN | WADT_STS |
| Internal Interrupt Error Handler (IEH) circuitry | No | Yes | IEH_SMI_EN | IEH_SMI_STS |

***Notes:***
1. The SCI_EN bit of the PM1_CNT register must be 1 to enable SCI. When SCI_EN is 0 AND GBL_SMI_EN = 1, SMI is enabled except for when an SCI is generated when the BIOS_RLS bit of the SMI_EN register is written to a 1 by BIOS.
2. When a power button override first occurs, the system transitions immediately to S5. The SCI will only occur after the next wake to S0 if the residual PWRBTNOR_STS status bit of the PM1_EN_STS register is not cleared prior to setting SCI_EN.
3. Refer to Chapter 26, "Customer General-Purpose I/O (GPIO)" for specific Customer GPIOs enabled for SCIs and/or SMIs.
4. GBL_STS being set will cause an SCI, even if the SCI_EN bit of the PM1_CNT register is not set. Software and BIOS must take great care not to set the BIOS_RLS bit of the SMI_EN register (which causes GBL_STS to be set) if the SCI handler is not in place.

### 16.7.1 PCI Express* SCI

PCI Express ports and the processor have the ability to cause a Power Management Event (PME) using messages. When a PME message is received, the SoC sets the PCI Express Status (PCI_EXP_STS) bit of the GPE0_STS_127_96 register. If the SCI Enable (SCI_EN) bit of the Power Management 1 Control (PM1_CNT) register and the PCI Express Enable (PCI_EXP_EN) bit of the GPE0_EN_127_96 register are also set, the SoC generates cause an SCI.

## 16.8 C States

The SoC processor power management circuitry, rather than the PMC, controls the C States. See Section 34.9, "Processor Power States - C-States" on page 913.

## 16.9 Dynamic LPC 24-MHz Clock Control

See Section 20.3.10, "Dynamic LPC Clock Control" on page 629.

## 16.10 Sleep States

For an overview of the Global Power States and the associated Sleep States see Section 34.8, "System Global Power States" on page 911. Power-supply cycling requirements associated with state changes are described in Chapter 33, "Resets and Voltage Sequencing".

### 16.10.1 Initiating State Changes when in the G0 (S0) Working State

These initiate a Global Power State change:

- In the G0 (S0) Working State, the Operating System Power Management (OSPM) software initiates the state change by following these steps:
  a. Mask interrupts
  b. Turn-off all bus-master enable bits
  c. Enabling the appropriate wake events that will trigger the exit from the selected state
  d. Set the desired sleep type in the three-bit Sleep Type (SLP_TYP) field of the Power Management 1 Control (PM1_CNT) register located in Host I/O Space at ABASE plus offset 4h. The SLP_TYP bits are in the RTC power well and are retained during G3 if the system has a functioning RTC battery.
  e. Set the Sleep Enable (SLP_EN) bit of the PM1_CNT register

  The hardware then attempts to gracefully put the system into the sleep state designated by SLP_TYP. Once in the sleep state, whenever one of the wake events enabled in "step c" occurs, the hardware transitions the system back to the G0 (S0) Working State which requires rebooting the system.
  The G3 Mechanical Off Global Power State cannot be entered by any operating-software mechanism. If the platform power is removed while in the G2 (S5) state, the G3 state is entered.
- In the G0 (S0) Working State, the user initiates the state change:
  a. The user presses the system Power button for less than 5.1 seconds. The SoC provides the active-low PMU_PWRBTN_N input signal pin to sense the position of the system Power button.
  b. The SoC then generates a power-management event using either a System Management Interrupt (SMI) or System Control Interrupt (SCI), to request the system's power-button driver to go to a sleep state.
  c. The OSPM sequence described in the bullet above this one is then followed. This is a controlled transition that allows OSPM to bring the system down in an orderly fashion by unloading applications, closing files, etc.
- In the G0 (S0) Working State, a Power Button Override is initiated by the user by pressing the system's Power button (active-low PMU_PWRBTN_N input signal pin) for more than 5.1 seconds in order to turn-off a hung system. This causes an unconditional, less graceful transition to the G2 (S5) Soft-Off State. Here the SoC first transitions directly to the early-boot phase and then transitions to the S5 Soft-Off state. Subsequent situations that can trigger a Warm Reset or Cold Reset are ignored.
  — The Intel® Management Engine and Innovation Engine firmware each has a mechanism to turn off a hung system similar to the Power-Button Override by writing bits in their power-management control registers. Subsequent situations that can trigger a Warm Reset or Cold Reset are ignored.

- In the G0 (S0) Working State, the SoC detects a Catastrophic Thermal Trip situation that may damage the SoC component. See Chapter 35, "THERMTRIP_N". The SoC asserts the THERMTRIP_N output signal pin. Except for the V3P30 voltage group, all SoC rails must shut down immediately. Just like a Power Button Override, Catastrophic Thermal Trip causes an unconditional, less graceful transition to the G2 (S5) Soft-Off State. Subsequent situations that can trigger a Warm Reset or Cold Reset are ignored.
  - If all platform power-supply sources to the SoC are shutdown at this point, and the platform doesn't have a working coin-cell battery, the SoC is incapable of retaining the cause of the global reset or straight-to-S5 during the subsequent reboot.
- If a reset button is part of the platform board design, and the user presses the button, it causes the PMU_RESETBUTTON_N SoC input signal pin to go active after a 16-ms de-bounce period). The SoC asserts the active low PMU_PLTRST_N output signal pin, and enters, or remains in, the G0 (S0) Working State.
  - If I/O Port CF9h (RST_CNT) register bit 3=0 in Host I/O Space, a Warm Reset occurs where no power cycles occur, and the SoC performs a system boot.
  - If I/O Port CF9h bit 3=1 in Host I/O Space, a Cold Reset occurs where the Core Power Well is cycled, and the SoC performs a system boot.
- Whenever a write of 06h is performed to I/O Port CF9h (RST_CNT) register, the SoC reacts as if the user pushed the system reset button. The SoC asserts the active low PMU_PLTRST_N output signal pin, and enters, or remains in, the G0 (S0) Working State.
  - If I/O Port CF9h bit 3=0 in Host I/O Space, a Warm Reset occurs where no power cycles occur, and the SoC performs a system boot.
  - If I/O Port CF9h bit 3=1 in Host I/O Space, a Cold Reset occurs where the Core Power Well is cycled, and the SoC performs a system boot.
- Whenever the system is shutdown to G2 (S5) or G3 by integrated manageability functions like Alert Standard Format (ASF).
- Whenever the TCO Watchdog Timer timeout occurs a second time within the SoC. If directed by the hard pin strap "No TCO Reboot Enable," the SoC asserts the active low PMU_PLTRST_N output signal pin, enters (or remains in) the G0 (S0) Working State. There is no power cycle involved and the destination is the G0 (S0) Working State. This is considered a "Warm Reset" and performs a system reboot. See Table 4-1, "SoC Hard Straps/Pin Straps and Descriptions".
- Whenever SoC hardware detects internal errors and is enabled to generated some type of reset. The types of reset and what happens is shown in Table 16-7, "Causes of Host Reset and Global Reset".

### 16.10.2 Initiating State Changes when in the G3 Mechanical-Off State

- When the SoC is in the G3 Mechanical Off state and the main system power is turned on, the platform board cycles-on the voltages it supplies to the SoC while systematically de-asserting the active-low RTEST_N, SRTCRST_N, and RSMRST_N input signals of the SoC. This initiates what is called a Cold Boot sequence and is shown in Figure 33-1, "ACPI Cold Boot Sequence". The SoC is put in the G2 (S5) Soft-Off state.
  - The SoC has circuitry to determine if this is the first time the system leaves the G3 Mechanical Off state. For this first-boot situation, the SoC uses the "AG3E Boot Select" Soft-Strap setting (see Chapter 4, "SoC Soft Straps") to determine whether to immediately transition to the G0 (S0) Working state and perform a "Cold Boot" or whether to wait in the G2 (S5) Soft-Off state for the power button or some other enabled wake event to transition to the G0 (S0) Working state.
  - During the first-boot sequence, the SoC retains the "AG3E Boot Select" Soft-Strap value in the AFTERG3_EN (AG3E) bit of the General PM Configuration B (GEN_PMCON_B) register for subsequent boot cycles. It retains the AG3E value until the SoC RTC power well loses power. BIOS can alter the AG3E register bit for subsequent boot cycles so that they behave differently than the first boot. If the AG3E bit is still set when the SoC transitions out of the G3 state, it not considered the first-boot situation. The retained value of AG3E is used instead of the "AG3E Boot Select" Soft Strap.

### 16.10.3 Returning to the S0 State when in the G2 (S5) Soft-Off State

The S5 Soft-Off sleep state is exited based on wake events. Before S5 was entered via the SLP_EN bit setting, software enabled the appropriate wake events in the Power Management 1 Enables and Status (PM1_EN_STS) register. When an enabled wake event occurs in S5, it forces the system to a full-on G2 (S0) Working state. Some non-critical subsystems, like rotational hard drives, might still be shut off and have to be brought back on using some other system-design mechanism.

Upon exit from the S5 Soft-Off state, the WAK_STS bit is set by the SoC. This bit is not affected by hard resets caused by a write to the Host I/O Port CF9h (RST_CNT) register, and is reset by writing a one to this bit, or by an internal power-good reset. The PM1_EN_STS register contains the wake event that caused the exit from the S5 state.

The possible causes of wake events (and their restrictions) are shown in Table 16-5 on page 537.

**Table 16-5. Power Management Wake Events (Sheet 1 of 2)**

| Cause of the Wake Event | How the Wake Event is Enabled | Wake Event Status Reported | Besides Wake from S5, can also Wake from Reset Types |
|---|---|---|---|
| RTC Alarm | RTC Alarm Enable (RTC_EN) | RTC Status (RTC_STS) | - |
| PMU_PWRBTN_N pin is active (Power Button) | Always enabled as a Wake Event. If an interrupt is desired, the Power Button Enable (PWRBTN_EN) bit must be set. | Power Button Status (PWRBTN_STS) | Notes 1, 2, 3 |
| Any GPIO signal pin that is configured and enabled for triggering a Wake Event | GPI General Purpose Events Enable (GPI_GPE_EN_*signal pin name)* See Chapter 8, "GPIO and PMC General Purpose Event (GPE)". | GPI General Purpose Events Status (GPI_GPE_STS_*signal pin name)* See Chapter 8, "GPIO and PMC General Purpose Event (GPE)". | - |

## Table 16-5. Power Management Wake Events (Sheet 2 of 2)

| Cause of the Wake Event | How the Wake Event is Enabled | Wake Event Status Reported | Besides Wake from S5, can also Wake from Reset Types |
|---|---|---|---|
| Integrated LAN Controllers | See Section 16.10.3.1, "LAN Power Management Wakeup" on page 539. | See Section 16.10.3.1, "LAN Power Management Wakeup" on page 539. | Notes 1, 2, 3 |
| Power Management Event (PME) from SoC integrated devices (PME_B0_STS) | See Section 16.10.3.2, "Other Internal Power Management Events" on page 540. | See Section 16.10.3.2, "Other Internal Power Management Events" on page 540. | - |
| PMU_WAKE_N pin is active (PCI Express WAKE# signal) | PCI Express Wake Disable (PCIEXP_WAKE_DIS) is not set. Modification of this bit has no impact on the value of the PCIEXP_WAKE_STS bit. | PCI Express Wake Status (PCIEXP_WAKE_STS) | - |
| SMB_LEG_ALRT_N pin is active (SMBus SMBALERT# Signal) | GPI General Purpose Events Enable (GPI_GPE_EN_SMB0_LEG_ALRT_N) | GPI General Purpose Events Status (GPI_GPE_STS_SMB0_LEG_ALRT_N) | - |
| Intel® Management Engine (Intel® ME) Non-Maskable Wake | Always enabled as a wake event. | ME_HOST_WAKE_STS (ME_HOST_WAKE_STS) | Notes 1, 2, 3 |
| Innovation Engine (IE) Non-Maskable Wake | Always enabled as a wake event. | IE Host Wake Status (IE_HOST_WAKE_STS) | Notes 1, 2, 3 |
| Wake Alarm Device | Wake Alarm Device Timer Enable (WADT_EN) | Wake Alarm Device Timer Status (WADT_STS) | - |

***Notes:***
1. Reset Type: The SoC transitions straight to the S5 Soft-Off state (not a gradual shutdown). Here the SoC first transitions to the beginning of the early-boot sequences and then transitions to the S5 Soft-Off state.
2. Reset Type: Host Reset with Power Down and the Host stays there when initiated by the Intel® ME.
3. Reset Type: Host Reset with Power Down and the Host stays there when initiated by the Innovation Engine.

### 16.10.3.1 LAN Power Management Wakeup

The integrated LAN controllers' advanced power management wakeup, or APM wakeup, was previously known as Wake on LAN (WoL). At power up, each SoC LAN controller reads the APM Enable bit from the LAN region of the SoC SPI Flash into the APM Enable (APME) bits of a controller register. This bit controls the enabling of LAN APM Wakeup. When APM Wakeup is enabled, the SoC LAN controller checks all incoming packets for Magic Packets. Each LAN controller can be configured to detect the APM Wakeup on one, both, or none if its two LAN ports. The LAN device driver has the ability to alter the APME bit after power up.

When all of its wakeup enables are in place the LAN controller sends a SoC-internal "enable" signal to the PMC. When the LAN controller detects a wake-up situation, it notifies the PMC through a SoC-internal "wakeup" signal. Before each time the PMC needs to transition from the G0 (S0) Working state to another sleep state, it senses the internal LAN enable signal to keep the LAN port powered-on during the new sleep state if LAN APM Wakeup is enabled. If for some reason the LAN controller is already powered-off during the G0 (S0) Working state, the internal enable signal has no effect on the LAN controller's power state.

If "WoL after SUS power loss" is configured in the LAN region of the SPI Flash, BIOS can use the WoL Enable Override (WOL_EN_OVRD) bit of the General PM Configuration B (GEN_PMCON_B) register to enable Wake-On-LAN regardless of the policies selected through the operating system. This bit is maintained in the RTC power well, therefore permitting WoL following a surprise power failure even in cases in which the system may have been running in S0 without the PME Enables set.

### 16.10.3.2 Other Internal Power Management Events

The SoC combines the LAN advanced power management enable and wakeup internal signals with the other SoC integrated PCI devices' power-management event enable and wakeup internal signals. The Power Management Event Bus 0 Status (PME_B0_STS) bit of the General Purpose Event 0 Status [127:96] (GPE0_STS_127_96) register is set to one by the SoC when any SoC-internal device with PCI Power Management capabilities on Host PCI Bus 0 asserts the equivalent of the PCI PME# signal. Additionally, if the PME_B0_EN bit of the GPE0_EN_127_96 register and SCI_EN bit in the PM1_CNT register are set, then the setting of the PME_B0_STS bit generates an SCI (or an SMI if SCI_EN is not set).

A Power Management Event (PME) from PM-enabled and powered-on SoC-internal devices that can set the PME_B0_STS bit are:

- Integrated LAN controllers (see Section 16.10.3.1, "LAN Power Management Wakeup" on page 539)
- Integrated SATA Controllers
- Integrated USB 2.0 Controllers
- Integrated USB 3.0 Controllers
- Intel® Management Engine (Intel® ME) Maskable Host Wake bit
- Innovation Engine (IE) Maskable Host Wake bit

If the PME_B0_EN bit is set, and the system is in the S3 or S4 state, or the S5 state due to SLP_TYP and SLP_EN, then the setting of the PME_B0_STS bit generates a wake event. If the system is in an S5 state due to power button override, then the PME_B0_STS bit does not cause a wake event or an SCI. PME_B0_STS is cleared by a software write of '1.'

*Note:* The S3 state for the SoC is not validated by Intel.

The truth table for situations when an integrated-device wakeup event is enable is shown in Table 16-6, "Power Management Event Enables". In the table:

— "Yes" indicated that the PME is enabled. Device power remains on during the S0-to-Sx transition.

— "No" indicates that the PME is not enabled. The power is turned off by the SoC to the affected devices during the S0-to-Sx transition. Exceptions to this are the Intel® ME and the IE in that they may stay powered-on (active) but the Sx PME wake might be disabled.

**Table 16-6. Power Management Event Enables**

| PME_B0_EN bit | PME_B0_S5_DIS bit | Internal Wakeup Enable signal from LAN | Wakeup and provide wake event status in S4? | Wakeup and provide wake event status in S5? |
|---|---|---|---|---|
| 0 | 0 | 0 | No | No |
| 0 | 1 | 0 | No | No |
| 0 | X | 1 | No | Yes, LAN only |
| 1 | 0 | 1 | Yes, all PME B0 sources including LAN | Yes, all PME B0 sources including LAN |
| 1 | 1 | 1 | Yes, all PME B0 sources including LAN | Yes, LAN only |
| 1 | 0 | 0 | Yes, all PME B0 sources including LAN | Yes, all PME B0 sources including LAN |
| 1 | 1 | 0 | Yes, all PME B0 sources including LAN | No |

## 16.10.4 Reset Triggers That Generate a State Change

Certain hardware situations generate a reset request to the PMC that can trigger the change of the Global Power State of the SoC. The SoC behavior after a reset trigger depends on the situation that generated the reset request.

See Section 16.13, "Reset Behavior" on page 543.

## 16.11 System Power Supplies, Planes, and Signals

All SoC power-control signal pins are described in Table 16-1, "Signal Names and Descriptions". The following is additional information about those signal pins that can be used for system power-supply control.

### 16.11.1 Platform Power Plane Control with PMU_SLP_S3_N and PMU_SLP_S45_N

When asserted, the active-low PMU_SLP_S3_N SoC output signal can be used to cut power to the system's VCC, VCCRAM, and VCCREF SoC power-supply pins. This same signal can be used to cut the termination voltage VTT to the DDR DIMMs. VTT is only delivered by the system to the DIMMs and not to the SoC.

When asserted, the active-low PMU_SLP_S45_N SoC output signal can be used to cut power to the system's VDDQ SoC power-supply pins. This is allowed because the SoC does not support the S3 sleep state.

Both of these pins can be configured as Customer GPIOs if not used in the system design.

### 16.11.2 COREPWROK SoC Input Signal

When asserted, the active-high COREPWROK signal is an indication to the SoC that the platform board Voltage Regulators (VRs) that supply the SoC Core Power Well rails have been stable for at least 10 ms. COREPWROK can be driven asynchronously. When COREPWROK is de-asserted, the SoC asynchronously asserts the active-low PMU_PLTRST_N signal. COREPWROK must not glitch, even if RSMRST_N is active (low)

The SoC output signal PMU_PLTRST_N de-asserts after the SoC input signal COREPWROK is asserted. Platforms which include a PCI Express* device may use the signal PMU_PLTRST_N to generate the PCI Express* signal PERST#. If the signal PMU_PLTRST_N is used to generate the PCI Express* signal PERST#, then the platform should not assert COREPWROK until at least 100ms after all the PCI Express power supplies and clocks have become stable and valid, as mandated within the PCIe Express* PCI Express* Card Electromechanical Specification.

### 16.11.3 SUSPWRDNACK SoC Output Signal

Suspend Power-Down Acknowledge (SUSPWRDNACK) is an active-high output signal that is asserted by the SoC when it does not require its Suspend (SUS) power well to be powered.

This pin can be configured as a Customer GPIO if not used in the system design.

## 16.12 Legacy Power Management Theory of Operation

Instead of relying on Advanced Configuration and Power Interface (ACPI) software, legacy power management uses BIOS and various hardware mechanisms. The scheme relies on the concept of detecting when individual subsystems are idle, detecting when the whole system is idle, and detecting when accesses are attempted to idle subsystems.

However, the operating system is assumed to be at least Advanced Power Management (APM) enabled. Without legacy APM calls, there is no quick way for the SoC to know when the system is idle (e.g., idle between keystrokes). The SoC does not support burst modes.

## 16.13 Reset Behavior

The SoC is divided into three reset partitions:

- Host Partition - The SoC hardware not including the Intel® Management Engine and the Innovation Engine.
- Intel® Management Engine Partition - Just the Intel® Management Engine portion of the SoC.
- Innovation Engine Partition - Just the Innovation Engine portion of the SoC.

The SoC can perform two main types of reset:

- Host Reset - Just the Host Partition hardware is reset. If a Host Reset is triggered and the PMC times out before receiving an internal acknowledge message from the processor, a Global Reset with power-cycle occurs.
- Global Reset - All three of the SoC partitions are reset and a power cycle is performed. During a Global Reset, all SoC functionality is reset except for information backed-up by the RTC Power Well or the SUS (Primary and DSW) Power Well. Also not reset are configuration information, and functional logic for controlling and reporting the reset. The Host, Intel® Management Engine, and Innovation Engine all power back up after the power cycle period.

### 16.13.1 Reset Request Conditions

Table 16-7, "Causes of Host Reset and Global Reset" shows the situations when the PMC is requested to perform a type of partition reset, possible power down/up cycle, or instead change (not gracefully) the Host sleep state to the G2 (S5) Soft-Off state and stay there until an enabled Wake Event occurs to return to the G0 (S0) Working state.

The latter is called "Straight to S5" and all SoC functionality is reset except for information backed-up by the RTC Power Well or the SUS Power Well. The terms "Primary" and "DSW" are used in the EDS register descriptions instead of "SUS." Also not reset are configuration information, and functional logic for controlling and reporting the reset. While Table 16-7, "Causes of Host Reset and Global Reset" shows Host Reset and Global Reset triggers, there are some situations where only the Intel® Management Engine Partition or Innovation Engine Partition is reset by the PMC.

**Table 16-7. Causes of Host Reset and Global Reset (Sheet 1 of 3)**

| Reset Request and Condition | Host Reset Without Power Cycle (Note 1) | Host Reset With Power Cycle (Note 2) | Global Reset With Power Cycle (Note 3) | Straight to S5 and Host Stays in S5 |
|---|---|---|---|---|
| **From power management state-change software** | | | | |
| The CPU is reset when going to the G2 (S5) Soft-Off state. SoC functionality that needs to remain operational or retain status during the Host S5 sleep state does not get reset in this case | - | Yes see Note 4 | see Note 5 | - |
| **From SoC input signal pins** | | | | |
| PMU_RESETBUTTON_N asserted and CF9h (RST_CNT Register) Bit 3 = 0 | Yes | - | see Note 5 | - |
| PMU_RESETBUTTON_N asserted and CF9h (RST_CNT Register) Bit 3 = 1 | - | Yes | see Note 5 | - |
| ADR_TRIGGER input signal is asserted when enabled to trigger a Global Reset. If the ADR flow is externally triggered, Global Reset will not be triggered. | - | - | Yes | - |
| Power Button (PMU_PWRBTN_N) 5.1-second override causes transition to S5 and reset asserts | - | - | - | Yes |
| **From software reset (I/O Port CF9h)** | | | | |
| Write of 06h to CF9h (RST_CNT Register) when CF9h Global Reset (CF9GR) bit = 0 in the ETR3 register of the PMC | Yes | - | see Note 5 | - |
| Write of 0Eh to CF9h (RST_CNT Register) when the SPD Reset (SPD_BIOS_RESET) bit = 1 in the PCH_PM_STS register | - | Yes see Note 6 | | - |
| Write of 0Eh to CF9h (RST_CNT Register) when CF9h Global Reset (CF9GR) bit = 0 in the ETR3 register of the PMC | - | Yes | see Note 5 | - |
| Write of 0Eh to CF9h (RST_CNT Register) when CF9h Global Reset (CF9GR) bit = 1 in the ETR3 register of the PMC | - | - | Yes | - |
| **From power failure and thermal alerts** | | | | |
| Power Failure: COREPWROK signal goes inactive in S0 | - | - | Yes | - |
| Processor Thermal Trip (THERMTRIP_N) causes transition to S5 and reset asserts | - | - | - | Yes |
| SoC internal thermal sensors signals a catastrophic temperature condition | - | - | - | Yes |

## Table 16-7. Causes of Host Reset and Global Reset (Sheet 2 of 3)

| Reset Request and Condition | Host Reset Without Power Cycle (Note 1) | Host Reset With Power Cycle (Note 2) | Global Reset With Power Cycle (Note 3) | Straight to S5 and Host Stays in S5 |
|---|---|---|---|---|
| **From detected hardware errors and interface time-out situations** | | | | |
| Total Cost of Ownership (TCO) Watchdog Timer reaches zero two times | Yes | - | see Note 5 | - |
| Host Partition Reset Entry Timeout (see Note 7) | - | - | Yes | - |
| PMC hardware detects an uncorrectable parity error on a data read from one of its SUS (Primary) Power Well registers | - | - | Yes | - |
| Power Management Watchdog Timer | - | - | - | Yes |
| PMC hardware detects that the system has been running too long without Intel® Management Engine firmware acknowledgment | - | - | - | Yes |
| **From shutdown cycles requested by the processor** | | | | |
| Special shutdown cycle from processor causes CF9h-like PMU_PLTRST_N when CF9h Global Reset (CF9GR) bit = 0 in the ETR3 register of the PMC and CF9h (RST_CNT Register) Bit 3 = 0 | Yes | - | see Note 5 | - |
| Special shutdown cycle from processor causes CF9h-like PMU_PLTRST_N when CF9h Global Reset (CF9GR) bit = 0 in the ETR3 register of the PMC and CF9h (RST_CNT Register) Bit 3 = 1 | - | Yes | see Note 5 | - |
| Special shutdown cycle from processor causes CF9h-like PMU_PLTRST_N when CF9h Global Reset (CF9GR) bit = 1 in the ETR3 register of the PMC | - | - | Yes | - |
| **From the Intel® Management Engine** | | | | |
| Intel® Management Engine-Triggered Host Reset without Power-Cycle | Yes | - | see Note 5 | - |
| Intel® Management Engine-Triggered Host Reset with Power-Cycle | - | Yes | see Note 5 | - |
| Intel® Management Engine-Triggered Host Reset with power down (host stays there) | - | Yes see Note 4 | see Note 5 | - |
| Intel® Management Engine promotes a Host Partition Reset to a Global Reset | - | - | Yes | - |
| Intel® Management Engine-Triggered Global Reset | - | - | Yes | - |
| Intel® Management Engine detects an unexpected Minute IA (MIA) error | - | - | - | Yes see Note 8 |
| Intel® Management Engine detects an unexpected Minute IA (MIA) shutdown error | - | - | - | Yes see Note 8 |
| Intel® Management Engine Hardware Uncorrectable Error | - | - | - | Yes see Note 8 |
| Intel® Management Engine Watchdog Timer Timeout | - | - | - | Yes see Note 8 |
| Intel® Management Engine-Triggered Power Button Override | - | - | - | Yes |
| **From the Innovation Engine** | | | | |
| Innovation Engine-Triggered Host Reset without Power-Cycle | Yes | - | see Note 5 | - |
| Innovation Engine-Triggered Host Reset with Power-Cycle | - | Yes | see Note 5 | - |
| Innovation Engine-Triggered Host Reset with power down (host stays there) | - | Yes see Note 4 | see Note 5 | - |
| Innovation Engine-Triggered Global Reset | - | - | Yes | - |

**Table 16-7. Causes of Host Reset and Global Reset (Sheet 3 of 3)**

| Reset Request and Condition | Host Reset Without Power Cycle (Note 1) | Host Reset With Power Cycle (Note 2) | Global Reset With Power Cycle (Note 3) | Straight to S5 and Host Stays in S5 |
|---|---|---|---|---|
| Innovation Engine detects an unexpected Minute IA (MIA) error | - | - | - | Yes see Note 9 |
| Innovation Engine detects an unexpected Minute IA (MIA) shutdown error | - | - | - | Yes see Note 9 |
| Innovation Engine Hardware Uncorrectable Error | - | - | - | Yes see Note 9 |
| Innovation Engine Watchdog Timer Timeout | - | - | - | Yes see Note 9 |
| Innovation Engine-Triggered Power Button Override | - | - | - | Yes |

***Notes:***
1. The SoC drops this type of reset request if received while the system is in the G2 (S5) Soft Off state.
2. The SoC does not drop this type of reset request if received while system is in a "software-entered" G2 (S5) Soft Off state. Here the SoC does perform the Host Reset with Power Cycle.
3. The SoC does not warn the processor. The Global Reset occurs without delay.
4. The PMC waits for an enabled Wake Event to occur before completing the Host Reset with Power Cycle.
5. If the PMC detects a problem while executing the Host Reset, the PMC promotes this request to be a Global Reset with Power Cycle.
6. SoC internal clocks are made ready for DIMM Serial Presence Detect (SPD) discovery.
7. Host Partition Reset Entry Timeout is automatically initiated if the hardware detects that the Platform Reset (PMU_PLTRST_N) sequence has not occurred within 5.1 seconds of being started.
8. Only the Intel® Management Engine is reset and not the Host or Innovation Engine
9. Only the Innovation Engine is reset and not the Host or Intel® Management Engine

## 16.14 Interaction with the Innovative Engine (IE)

The IE provides the fan control FAN_PWM[3:0] and FAN_TACH[3:0] signals that are listed in Chapter 37, "Innovation Engine".

A number of IE error conditions are reported internally to the PMC so that the PMC can take the appropriate action, typically a Global Reset. See Table 16-7, "Causes of Host Reset and Global Reset".

The IE and PMC also interact with the following:

- Power Up
- Power Down
- IE power control
- Expiration of the IE-based Anti-Theft Watchdog Timer

## 16.15 Asynchronous DRAM Refresh (ADR)

The ADR feature of the SoC provides a mechanism to enable preservation of key data in battery-backed DDR system memory, in the event of AC power-supply failure. For all processor cores of the product SKU, the SoC ADR sequence forces an L1 cache flush of "dirty write" data to the battery-backed, or supercapacitor-backed, DDR system memory DIMMs and then places DIMMs in the battery/supercapacitor-backed Self-Refresh Mode defined by JEDEC. Dirty writes located in the SoC internal memory data buffers are also flushed. For ADR to be successful, the platform board must sustain valid power levels to the SoC for a minimum of 50 µs once the SoC initiates and enters the ADR sequence.

As a configurable option, the SoC can also flush dirty-write data from the L2 caches to the battery-backed system-memory DIMMs. This adds a certain time to the process and the platform board must sustain valid power to the SoC during this period, too.

The SoC ADR_TRIGGER input signal is the early-warning alert asserted by the platform and system power supply indicating that AC power has been lost and that the platform voltage rails will soon be dropping. Before the platform-board circuitry asserts RSMRST_N or deasserts COREPWROK, the ADR_TRIGGER signal must be asserted and trigger the ADR flow of the SoC.

Typically, proper fencing protocol before ADR triggers is not needed if both L1 and L2 caches are flushed. This may not be true if the platform design contains external DMA subsystems or if multiple SoC nodes mirror data using an external Non-Transparent Bridge (NTB). The discussion of such fencing techniques is beyond the scope of this document.

*Note:* An SoC internal register bit is used to convey to BIOS that a successful ADR sequence occurred before the most-recent loss of power or Global Reset. For the ADR feature to function properly, a battery is required to keep the RTC power well powered when in the G3 state.

### 16.15.1 Configuring ADR

To provide imminent power loss as a condition to initiate ADR, BIOS ensures that the SoC pin-function MUXing is configured so that the SoC provides the ADR_TRIGGER and ADR_COMPLETE signal pins for communication with the platform board circuitry. By default, these signal pins are Customer GPIO output signals GPIO[81] and GPIO[0] respectively.

*Note:* The SoC ensures that the ADR_COMPLETE output signal is glitch-free, even during the SUS-Well power ramp.

The SoC ADR mechanism is configured and controlled by the 32-bit ADR Enable (ADR_EN) register. The register is one of the PMC Memory Mapped Registers located in Host Memory Space at the address specified by the Power Management Base (PWRMBASE) register, plus offset F0h. The PWRMBASE register is located in Configuration Space at Bus 0, Device 31 (decimal), Function 2, offset 48h.

*Note:* The PWRM Enable (PWRM_EN) bit must be set to enable the decode of the memory-mapped address range pointed by PWRMBASE. PWRM_EN is located in the 32-bit ACPI Control (ACTL) register which is located in Configuration Space at Bus 0, Device 31 (decimal), Function 2, offset 44h.

The ADR feature is enabled/disabled by the ADR Feature Enable (ADR_FEAT_EN) bit (bit 0) of the ADR_EN register. When the feature is enabled, the ADR_TRIGGER signal pin, indicating an impending power loss detected by the board, is used to start the ADR process. Any of a number of "Qualified Internal Events" can also start the ADR process provided they are enabled in the ADR_EN register.

Bit 29 of the ADR_EN register enables/disables whether a Global Reset is generated after the SoC internal-data flushing completes.

BIOS must indicate whether the L2 Cache is to be flushed or not flushed during the ADR process. BIOS does this by programming the L2_FLUSH bit (bit 0) of the P_CR_ADR_COMMAND_0_0_0_MCHBAR Register located in Host Root Memory Space at MCHBAR, offset 712Ch. MCHBAR is defined by the two 32-bit registers MCHBAR_LO_0_0_0_PCI and MCHBAR_HI_0_0_0_PCI located at offsets 48h and 4Ch in configuration space for the System Agent (Bus 0, Device 0, Function 0).

- L2_FLUSH bit = 0: The SoC does not flush L2 Cache during the ADR flow.
- L2_FLUSH bit = 1: The SoC flushes L2 Cache on all processor modules as part of the ADR flow when it is initiated by the ADR_TRIGGER signal pin.

#### 16.15.1.1 Qualified Internal Events

There are a number of SoC-internal qualified events that can be enabled in the ADR_EN register to trigger the ADR sequence and generate a Global Reset after the internal cache-flush sequence completes. These are not associated with the Qualified External Event where the assertion of the ADR_TRIGGER signal pin by the platform board triggers the ADR sequence. The Qualified Internal Events include:

- Power Button Override
- Intel® ME-Initiated Power Button Override
- IE-Initiated Power Button Override
- SoC PMC Firmware Watchdog Timer
- Intel® ME Firmware Watchdog Timer
- IE Firmware Watchdog Timer
- SoC PMC Firmware-Initiated Global Reset
- Intel® ME-Initiated Global Reset
- IE-Initiated Global Reset
- SoC PMC SUS RAM Parity Error
- SYS_PWROK Failure
- Intel® ME Uncorrectable Error
- IE Uncorrectable Error

## 16.15.2 Configuring the Copy-to-Flash Extension

The Copy-to-Flash (C2F) extension to the ADR sequence is a special platform-level and BIOS sequence. It can extend the system's sequence of events when the SoC ADR sequence is triggered by the platform (ADR_TRIGGER signal for imminent power loss).

In some system designs, the information retained in the system memory after the ADR sequence (valid SDRAM power and SDRAMs in Self-Refresh Mode) is considered volatile. Such system designs require all or some of the system memory to also be written to persistent system memory such as a Solid-State Device (SSD).

To do this, the system platform needs to restore system power for a period of time, usually with some kind of backup power source or supercapacitor for the system, and have the SoC firmware/software write information to the persistent memory subsystem. Also, the platform may require special, on-board registers that are attached to the configured SoC Customer GPIO input signal pins, that later can be read by BIOS to determine a C2F situation.

Normally, when ADR_COMPLETE is asserted by the platform, the ADR sequence is not followed by a Global Reset. To facilitate C2F, the SoC can be configured to trigger a Global Reset once ADR_COMPLETE is asserted. To do so, these conditions must be met:

- ADR_GPIO_RST_EN bit of the ADR_EN register = 1, and
- ADR_FEAT_EN bit of the ADR_EN register = 1, and
- The product SKU supports the ADR feature, and
- The SoC internal ADR sequence has completed.

## 16.15.3 NVDIMM Copy-to-Flash

If NVDIMM is used for DRAM system memory by the platform design, the NVDIMMs integrated save-to-flash and restore-from-flash operations can be used once ADR_COMPLETE is asserted after an ADR sequence. NVDIMM has a SAVE# input pin, or a SAVE Serial Presence Detect (SPD) command via its SMBus interface, to instruct it to save its DRAM information (now in Self-Refresh Mode and powered by a resident supercapacitor) to its Flash memory. They also have a pin or SPD command to for the platform or BIOS to perform a RESTORE once the system returns to normal operation.

*Note:* At the time of publication of this document, there is no industry standard for NVDIMMs and for their SAVE and RESTORE pins and/or SPD commands. NVDIMM products typically follow the RDIMM JEDEC standards which has two reserved DIMM pins and reserved SPD registers. Platform designers and BIOS developers need to refer to NVDIMM technical information from NVDIMM manufactures for implementation details.

### 16.15.4 ADR Sequence of Events

1. BIOS has previously configured the SoC and has made necessary adjustments to the ADR_EN register and then sets the ADR_FEAT_EN bit. The ADR mechanism is armed and waiting.
2. The ADR_TRIGGER signal is asserted by the platform board indicating an imminent power failure has been detected or one of the enabled Qualified Internal Events (within the SoC) occurs. While the Global Reset triggers are captured in the GBLRST_CAUSE0 and GBLRST_CAUSE1 memory-mapped registers, the ADR sequence starts only if the particular trigger is enabled in the ADR_EN register.

*Note:* The active-high ADR_TRIGGER SoC pin is an open-drain output signal as well as an input signal. Not only can the board drive this signal pin, the SoC drives this pin high when one of the enabled Qualified Internal Events occurs. As long as the ADR-triggered event is valid, the ADR_TRIGGER remains asserted (logic high) by the SoC.

***Warning:*** To indicate the deasserted state, the platform (and SoC) must float (High-Z state) the ADR_TRIGGER signal pin. The platform must have a pull-down resistor tied to this pin. The platform board must only drive this pin to logic 1, when it intends to assert ADR_TRIGGER. It must never drive this pin to a logic 0. The SoC drives this pin to a logic 1 only when one of the enabled, Qualified Internal Events occur. Otherwise the SoC floats (Hi-Z) this pin. The SoC never drives it to a logic 0.

3. The SoC closes all access paths to memory space.
4. All cores are put to sleep and L1 cache flushes are performed.
5. Internal data buffers holding dirty memory-write information are flushed.
6. Once all cores are idle, DRAM throttling is disabled.
7. If the SoC is configured to also flush the L2 caches during ADR, dirty writes in L2 cache are cleaned (written to DRAM). This could take some time to perform.
8. The SoC System Agent circuitry is made idle.
9. The SoC puts the SDRAM components in Self-Refresh Mode.
   - It is the platform's responsibility to maintain SDRAM/DIMM power and maintain the SDRAM components' Clock Enable input signals during Self-Refresh mode.
   - When the SDRAM is on an NVDIMM product, a supercapacitor, that is part of the NVDIMM product, powers the SDRAM while in Self-Refresh Mode.

*Note:* In order to continue the ADR sequence, the SoC internal timer for ADR completion must not have expired at this point. Ending the sequence at this point indicates that there was not enough time or power-level to complete the ADR sequence or that an internal error that prevented completion of the sequence. Nothing happens and the G3 state (mechanical off) is entered when power is finally lost.

10. The SoC sets ADR_RST_STS, bit 16 of the PCH Power Management Status (PCH_PM_STS) register located in memory space at PWRMBASE, offset 1Ch.
    — The ADR_RST_STS register bit is powered by the SoC RTC Power Well. The SoC must be provided power for the RTC well (VCCRTC_3P3_G3, pin AU48) for this register bit to be preserved during the G3 State (Mechanical Off).
11. The SoC asserts the ADR_COMPLETE output signal pin.
12. Depending on the information in the ADR_EN register, the SoC does one of the following:

- Initiates a Global Reset with an automatic power-up to the S0 state and start of BIOS execution. The platform-defined Copy-to-Flash extension relies on this reset and BIOS entry.
  — Before the SoC starts the power-up, it waits in the S5 state for about five seconds. No wake event is needed to exit S5.
  — Internal SoC registers are reset except for the RTC power-well registers and registers and functional logic in other SoC power wells needed to report this Global Reset.
- Transitions directly to the S5 State and waits for some kind of wake event.
  — The SoC output signal pins PMU_SLP_S3_N and PMU_SLP_S45_N are asserted.
  — Internal SoC registers are reset except for the RTC power-well registers and registers and functional logic in other SoC power wells needed to report this status.
  — Once the SoC asserts the ADR_COMPLETE output signal pin, and the "AG3E Boot Select" Soft Strap = 1 (see Section 4.2, "SoC Soft Straps" on page 199), the platform needs to drive the PMU_PWRBTN_N or PMU_WAKE_N SoC input signal, or configure and drive a wake-supported Customer GPIO (see Chapter 26, "Customer General-Purpose I/O (GPIO)") input to wake-up the SoC from the S5 state.
- Once awakened, the SoC powers-up and enters the S0 state where BIOS begins instruction execution.
  — If the "AG3E Boot Select" Soft Strap = 0, the SoC does not wait for a wake-up event to proceed. Here the SoC enters the S0 state and BIOS commences. For Soft Strap details, see Chapter 4, "Strapping and Configuration".

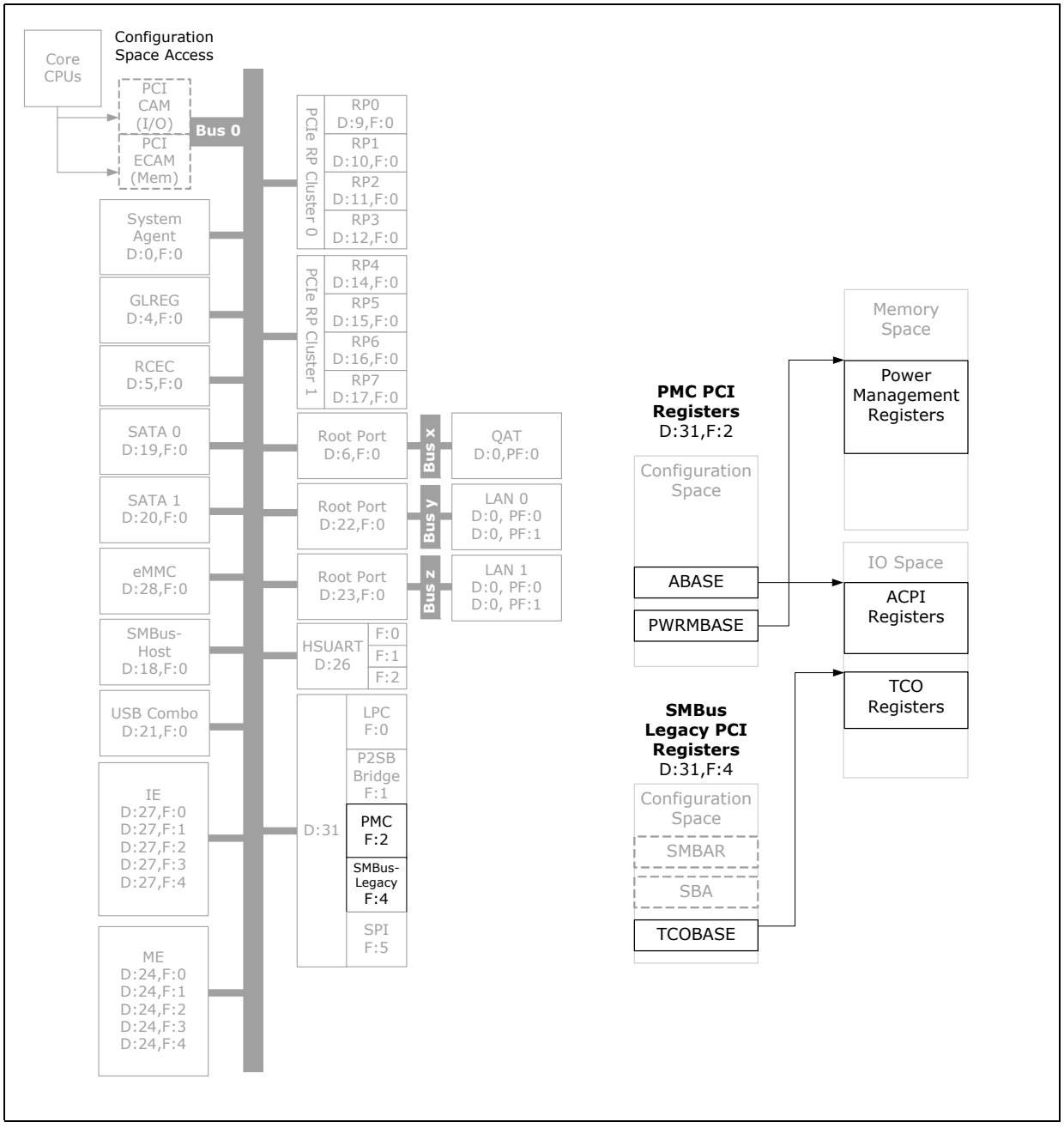
### 16.15.5 BIOS after Successful ADR

BIOS uses the ADR_RST_STS, bit 16 of the PCH Power Management Status (PCH_PM_STS) register to determine that a successful ADR sequence occurred before BIOS initiated. It also uses the SoC registers of the Customer GPIO input signals provided by the platform to signify a successful Copy-to-Flash (C2F). The BIOS sequences for ADR and C2F are not discussed in this document.

## 16.16 Register Map

Figure 16-2, "Register Map" shows the associated registers from a system software viewpoint.

**Figure 16-2. Register Map**



## 16.16.1 PCI Configuration and Capabilities

### Table 16-8. Configuration and Capabilities Register Map

| Offset from Configuration Space of B0:D31:F2 (hex) | Register Short Name | Description |
|---|---|---|
| 00 | PCIID | PCI Identifier |
| 04 | PCISTS_PCICMD | Device Status & Command |
| 08 | PCICC_PCIRID | Class Code & Revision ID |
| 0C | PCIHTYPE_PCIMLT | Header Type and Master Latency Timer |
| 10 | PM_DATA_BAR | 32-bit Power Management Data Base Address Register |
| 2C | PCISID | Subsystem Identifiers |
| 40 | ABASE | ACPI Base Address |
| 44 | ACTL | ACPI Control |
| 48 | PWRMBASE | PM Base Address |
| A0 | GEN_PMCON_A | General PM Configuration A |
| A4 | GEN_PMCON_B | General PM Configuration B |
| A8 | BM_CX_CNF | BM_BREAK_EN and Cx State Configuration Register |
| AC | ETR3 | Extended Test Mode Register 3 |
| B0 | PMC_THROT_1 | PMC Throttling 1 |
| C0 | MDAP | Maximum DMA Alignment Period |
| F8 | MANID | Manufacturer ID |

## 16.16.2 Memory-Mapped Registers

### Table 16-9. Relocatable Memory-Mapped Registers

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| PWRMBASE | 1C | PCH_PM_STS | PCH Power Management Status |
| PWRMBASE | F0 | ADR_EN | ADR Enable |
| PWRMBASE | 124 | GBLRST_CAUSE0 | Global Reset Causes |
| PWRMBASE | 128 | GBLRST_CAUSE1 | Global Reset Causes |
| MCHBAR_LO_0_0_0_PCI<br>MCHBAR_HI_0_0_0_PCI | 712h | P_CR_ADR_COMMAND_0_0_0_MCHBAR | ADR Command |

The PMC device/function has no Fixed-Address Memory-Mapped Registers.

## 16.16.3 I/O-Mapped Registers

### Table 16-10. Relocatable I/O-Mapped Registers - Power Management Controller (PMC)

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| **ACPI Base Address (ABASE)** | | | |
| ABASE | 00 | PM1_EN_STS | Power Management 1 Enables and Status |
| ABASE | 04 | PM1_CNT | Power Management 1 Control |
| ABASE | 08 | PM1_TMR | Power Management 1 Timer |
| ABASE | 30 | SMI_EN | SMI Control and Enable |
| ABASE | 34 | SMI_STS | SMI Status Register |
| ABASE | 40 | GPE_CTRL | General Purpose Event Control |
| ABASE | 44 | DEVTRAP_STS | Device Trap Status Register |
| ABASE | 50 | PM2A_CNT_BLK | PM2a Control Block |
| ABASE | 54 | OC_WDT_CTL | Over-Clocking WDT Control |
| ABASE | 80 | GPE0_STS_31_0 | General Purpose Event 0 Status [31:0] |
| ABASE | 84 | GPE0_STS_63_32 | General Purpose Event 0 Status [63:32] |
| ABASE | 88 | GPE0_STS_95_64 | General Purpose Event 0 Status [95:64] |
| ABASE | 8C | GPE0_STS_127_96 | General Purpose Event 0 Status [127:96] |
| ABASE | 90 | GPE0_EN_31_0 | General Purpose Event 0 Enable [31:0] |
| ABASE | 94 | GPE0_EN_63_32 | General Purpose Event 0 Enable [63:32] |
| ABASE | 98 | GPE0_EN_95_64 | General Purpose Event 0 Enable [95:64] |
| ABASE | 9C | GPE0_EN_127_96 | General Purpose Event 0 Enable [127:96] |

### Table 16-11. Fixed-Address I/O-Mapped Registers

| I/O Address (hex) | Register Short Name | Register Long Name |
|---|---|---|
| CF9 | RST_CNT | Reset Control Register |

### Table 16-12. Relocatable I/O-Mapped Registers - Legacy SMBus Controller

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| TCOBASE | 00 | TRLD | TCO_RLD Register |
| TCOBASE | 02 | TDI | TCO_DAT_IN Register |
| TCOBASE | 03 | TDO | TCO_DAT_OUT Register |
| TCOBASE | 04 | TSTS1 | TCO1_STS Register |
| TCOBASE | 06 | TSTS2 | TCO2_STS Register |
| TCOBASE | 08 | TCTL1 | TCO1_CNT Register |
| TCOBASE | 0A | TCTL2 | TCO2_CNT Register |
| TCOBASE | 0C | TMSG | TCO_MESSAGE1 and TCO_MESSAGE2 |
| TCOBASE | 0E | TWDS | TCO_WDSTATUS Register |
| TCOBASE | 0F | RSV_1 | Reserved_1 |
| TCOBASE | 10 | LE | LEGACY_ELIM Register |
| TCOBASE | 11 | RSV_2 | Reserved_2 |
| TCOBASE | 12 | TTMR | TCO_TMR Register |
| TCOBASE | 14 | RSV_3 | Reserved_3 |
| TCOBASE | 1C | RSV_4 | Reserved_4 |

## 16.16.4 Sideband Registers

### Table 16-13. Sideband Registers - Power Management Control (PMC)

| Sideband Port (hex) | Offset (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| 82 | 82_00B0 | APM_STS_CNT | Advanced Power Management Status and Control Port |
| | 82_00B4 | MISC_SBR0 | Misc SB Reg0 |
| | 82_00B8 | MISC_SBR1 | Misc SB Reg1 |
| | 82_00BC | MISC_SBR2 | Misc SB Reg2 |

### Table 16-14. Sideband Registers - Legacy SMBus (for TCO)

| Sideband Port (hex) | Offset (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| CF | CF_0000 | TCOCFG | TCO Configuration |

§ §

# 17 Embedded Multi-Media Card (eMMC)

The Embedded Multi-Media Controller (eMMC) is a high-speed Flash memory controller. The controller presents itself to the host as Bus 0, Device 28, and Function 0 during PCI bus enumeration process. This controller handles eMMC protocol at transmission, packing data, adding cyclic redundancy check (CRC), start/end bit, and checks for transaction format correctness.

**Figure 17-1. What is Covered in This Chapter**

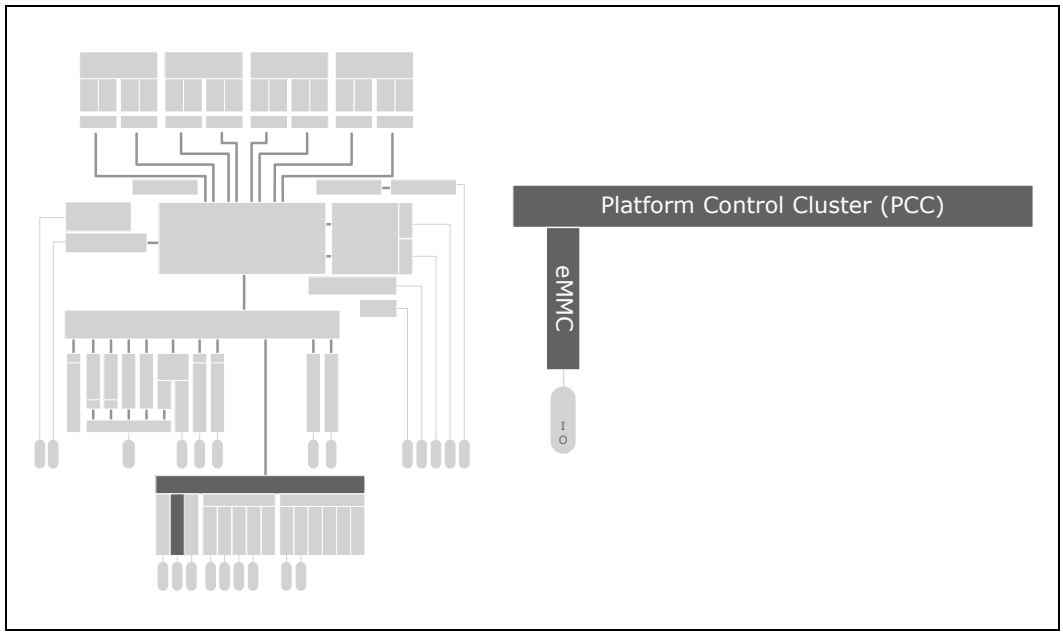



**Table 17-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| eMMC Spec | 5.0 | Oct 2013 | *Embedded Multi-Media Card (eMMC), Electrical Standard* |

## 17.1 Signal Descriptions

The signal descriptions are shown in Table 17-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing". The Direction column of Table 17-2 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit

**Table 17-2. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| EMMC_CLK | I,O | Yes | **Clock** |
| EMMC_CMD | I,O | Yes | **Command:** The platform requires a 20k Ohm pull-up resistor, tied to 1.8V for this signal. |
| EMMC_D[7:0] | I,O | Yes | **Data:** The platform requires a 20k Ohm pull-up resistor, tied to 1.8V for these signals. |
| EMMC_IRCOMP | I,O | No | **RCOMP for the eMMC interface**: The platform board must provide a 200 Ω, 1% resistor connected from this pin to VSS. |
| EMMC_STROBE | I | Yes | **Strobe for Rx Data** |

## 17.2 Feature List

- Supports eMMC 5.0 and eMMC 4.5.
- 8-bit interface, maximum frequency of 200 MHz (data transfer rates of 400 MB/sec).
- Supports interrupt coalescing.
- Supports both DMA and non-DMA mode operation.
- Supports data transfers in 1-, 4- and 8-bit mode.
- Supports cyclic redundancy check CRC7 for command and CRC16 for data integrity.
- eMMC interface can be a boot interface to boot an Operating System.

## 17.3 Architectural Overview

There are two supported modes of operation, PCI and ACPI modes. Upon controller power-on this controller exposes itself as a PCI-compliant device to the host space. For details on how to configure this device as a ACPI device, see Section 17.4, "Address Map" on page 559.

### 17.3.1 eMMC Mass Storage Command and Data Flows

Host software is to manage data transfer to and from controller to the host memory.

Following steps describe data flows to and from the controller.

1. Host prepares the controller for data transfer by writing DMA descriptor(s) in the host memory.
2. Host write the controller registers to initiate the descriptor chain.
3. Controller moves data from the host memory to the device card.
4. Controller DMA notifies the IOxAPIC after the DMA is done.
5. IOxAPIC forwards the interrupt the host as MSI.

### 17.3.2 PCI Mode and ACPI Mode

The eMMC can be configured to work in either PCI Mode or ACPI Mode. The configuration is done via a write to the IOSF2OCP Sideband register. By default the eMMC is set to PCI Mode. To change the eMMC to ACPI Mode, it first needs to be enumerated and assigned a BAR. Once this is done, it is possible to configure the device to be an ACPI device (ACPI Mode). Its address shall be the configured BAR address.

## 17.4 Address Map

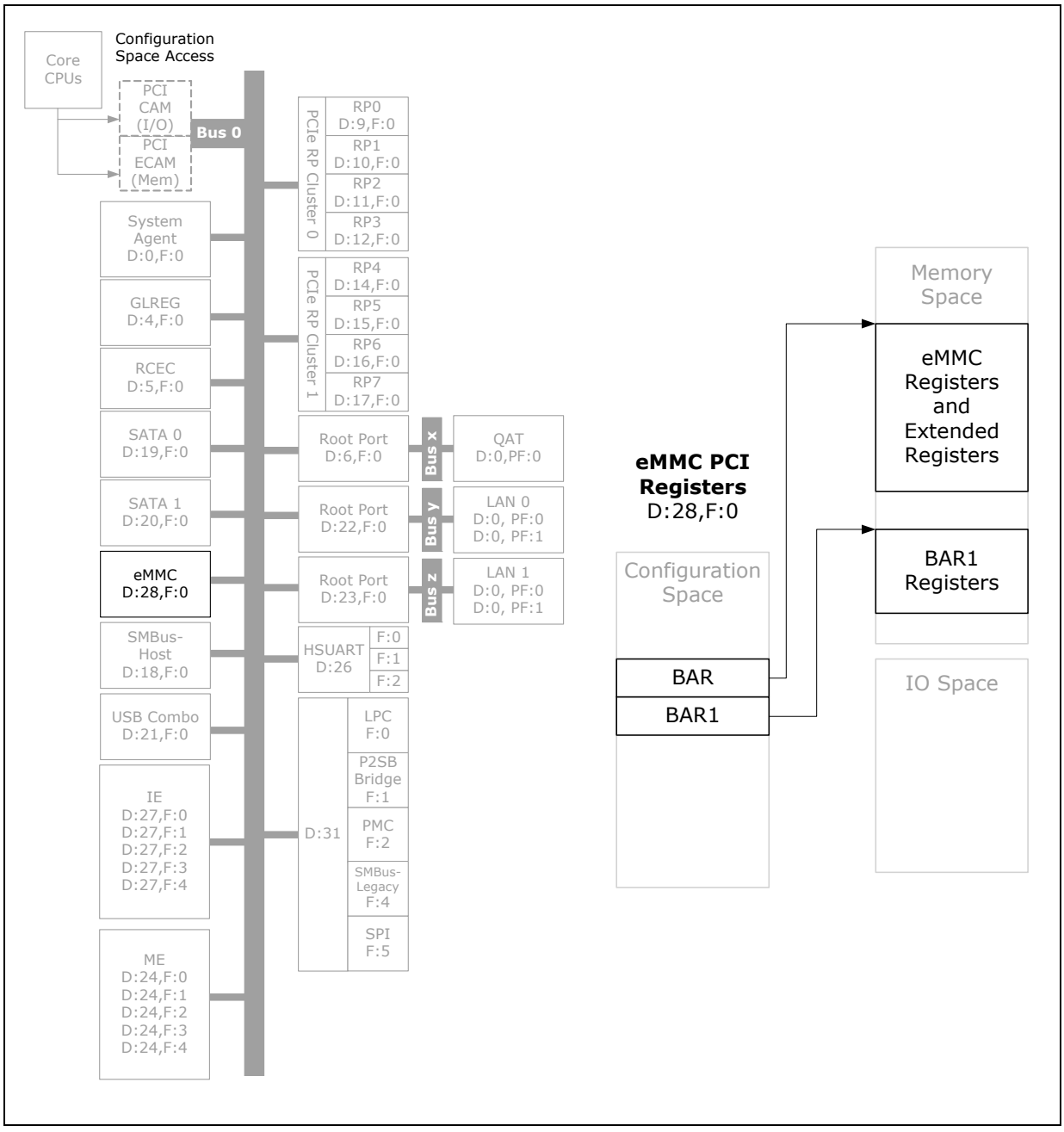
From the host point of view, the eMMC device has a 4KB BAR address space. The 4KB is divided into two regions

1. The device internal registers – a 2KB address space which maps the device internal registers.
2. Extended Configuration Registers – a 2KB address space which maps all the extended configuration.

## 17.5 Register Map

Figure 17-2 shows the associated registers from a system software viewpoint.

**Figure 17-2. Register Map**



## 17.5.1 PCI Configuration and Capabilities

### Table 17-3. Configuration and Capabilities Register Map

| Address | Default | Instance Name | Name |
|---|---|---|---|
| 0x0 | 19DB_8086h | DEVVENDID | Device ID and Vendor ID Register |
| 0x4 | 0010_0000h | STATUSCOMMAND | Status and Command |
| 0x8 | 0805_0100h | REVCLASSCODE | Revision ID and Class Code |
| 0xC | 0080_0000h | CLLATHEADERBIST | Cache Line Latency Header and BIST |
| 0x10 | 0000_0004h | BAR | Base Address Register |
| 0x14 | 0000_0000h | BAR_HIGH | Base Address Register High |
| 0x18 | 0000_0004h | BAR1 | Base Address Register1 |
| 0x1C | 0000_0000h | BAR1_HIGH | Base Address Register1 High |
| 0x2C | 0000_0000h | SUBSYSTEMID | Subsystem Vendor and Subsystem ID |
| 0x30 | 0000_0000h | EXPANSION_ROM_BASEADDR | EXPANSION ROM Base Address |
| 0x34 | 0000_0080h | CAPABILITYPTR | Capabilities Register |
| 0x3C | 0000_0000h | INTERRUPTREG | Interrupt Register |
| 0x80 | 0003_9001h | POWERCAPID | Power Management Capability ID |
| 0x84 | 0000_0008h | PMECTRLSTATUS | Power Management Control Status |
| 0x90 | F014_0009h | PCIDEVIDLE_CAP_RECORD | PCI Device Idle Capability Record |
| 0x94 | 0140_0010h | DEVID_VEND_SPECIFIC_REG | DEVID Vendor Specific Reg |
| 0x98 | 0000_8041h | D0I3_CONTROL_SW_LTR_MMIO_REG | SW LTR Update MMIO Location Register |
| 0x9C | 0000_81C1h | DEVICE_IDLE_POINTER_REG | Device IDLE Pointer Register |
| 0xA0 | 0029_0800h | D0I3_MAX_POW_LAT_PG_CONFIG | Device PG Config |
| 0xB0 | 0000_0000h | GEN_REGRW1 | General Purpose Read Write Register1 |
| 0xB4 | 0000_0000h | GEN_REGRW2 | General Purpose Read Write Register2 |
| 0xB8 | 0000_0000h | GEN_REGRW3 | General Purpose Read Write Register3 |
| 0xBC | 0000_0000h | GEN_REGRW4 | General Purpose Read Write Register4 |
| 0xC0 | 0000_0000h | GEN_INPUT_REG | General Purpose Input Register |
| 0xF8 | 0000_000Fh | MANID | Manufacturers ID |

## 17.5.2 Memory-Mapped Registers

### Table 17-4. Memory-Mapped Registers (Sheet 1 of 2)

| Address | Default | Instance Name | Register Long Name |
|---|---|---|---|
| 0x0 | 0000_0000h | SDMASYSADDR | SDMA System Address Register/Argument2 Register |
| 0x4 | 0000h | BLOCKSIZE | BlockSize Register |
| 0x6 | 0000h | BLOCKCOUNT | BlockCount Register |
| 0x8 | 0000_0000h | ARGUMENT1 | Argument1 Register |
| 0xC | 0000h | TRANSFERMODE | TransferMode Register |
| 0xE | 0000h | COMMAND | Command Register |
| 0x10 | 0000_0000h | RESPONSE1 | Response Register |
| 0x14 | 0000h | RESPONSE2 | Response Register |
| 0x16 | 0000h | RESPONSE3 | Response Register |
| 0x18 | 0000h | RESPONSE4 | Response Register |
| 0x1A | 0000h | RESPONSE5 | Response Register |
| 0x1C | 0000h | RESPONSE6 | Response Register |
| 0x1E | 0000h | RESPONSE7 | Response Register |
| 0x20 | 0000_0000h | DATAPORT | Buffer DataPort Register |
| 0x24 | 0000_0000h | PRESENTSTATE | PresentState Register |
| 0x28 | 00h | HOSTCONTROL1 | HostControl1 Register |
| 0x29 | 00h | POWERCONTROL | PowerControl Register |
| 0x2A | 80h | BLOCKGAPCONTROL | BlockGapControl Register |
| 0x2B | 00h | WAKEUPCONTROL | Wakeup Control Register |
| 0x2C | 0000h | CLOCKCONTROL | Clock Control Register |
| 0x2E | 00h | TIMEOUTCONTROL | Timeout Control Register |
| 0x2F | 00h | SOFTWARERESET | Software Reset Register |
| 0x30 | 0000h | NORMALINTRSTS | Normal Interrupt Status Register |
| 0x32 | 0000h | ERRORINTRSTS | ErrorInterruptStatus_Register |
| 0x34 | 0000h | NORMALINTRSTSENA | Normal Interrupt Status Enable Register |
| 0x36 | 0000h | ERRORINTRSTSENA | Error Interrupt Status Enable Register |
| 0x38 | 0000h | NORMALINTRSIGENA | Normal Interrupt Signal Enable Register |
| 0x3A | 0000h | ERRORINTRSIGENA | Error Interrupt Signal Enable Register |
| 0x3C | 0000h | AUTOCMDERRSTS | Auto CMD12 Error Status Register |
| 0x3E | 0000h | HOSTCONTROL2 | Host Control2 Register |
| 0x40 | 0000_0000_0000_0000h | CAPABILITIES | Capabilities Register |
| 0x48 | 0000_0000_0000_0000h | MAXCURRENTCAP | Maximum Current Capabilities Register |
| 0x50 | 0000h | FORCEEVENTFORAUTOCMDERRORSTATUS | Force Event REGISTER for AUTO CMD Error Status |
| 0x52 | 0000h | FORCEEVENTFORERRINTSTS | Force Event Register for Error Interrupt Status |
| 0x54 | 00h | ADMAERRSTS | ADMA Error Status Register |
| 0x58 | 0000_0000h | ADMASYSADDR01 | ADMA System Address Register0&1 |
| 0x5C | 0000h | ADMASYSADDR2 | ADMA System Address Register1 |

## Table 17-4. Memory-Mapped Registers (Sheet 2 of 2)

| | | | |
|---|---|---|---|
| 0x5E | 0000h | ADMASYSADDR3 | ADMA System Address Register1 |
| 0x60 | 0003h | PRESETVALUE0 | Preset Value Register for Initialization |
| 0x62 | 0003h | PRESETVALUE1 | Preset Value Register for Default Speed |
| 0x64 | 0003h | PRESETVALUE2 | Preset Value Register for High Speed |
| 0x66 | 0003h | PRESETVALUE3 | Preset Value Register for SDR12 |
| 0x68 | 0002h | PRESETVALUE4 | Preset Value Register for SDR25 |
| 0x6A | 0001h | PRESETVALUE5 | Preset Value Register for SDR50 |
| 0x6C | 0000h | PRESETVALUE6 | Preset Value Register for SDR104 |
| 0x6E | 0002h | PRESETVALUE7 | Preset Value Register for DDR50 |
| 0x70 | 0000_0000h | BOOTTIMEOUTCNT | Boot Timeout Control Register |
| 0x74 | 0000h | PRESETVALUE8 | Preset Value Register for DDR50 |
| 0xFC | 0000h | SLOTINTRSTS | Slot Interrupt Status Register |
| 0xFE | 1002h | HOSTCONTROLLERVER | Host Controller Version Register |

### 17.5.3 I/O-Mapped Registers

There are no I/O mapped registers.

§ §

# 18 High-Speed UART Controller

The SoC contains four[1] integrated High-Speed Universal Asynchronous Receiver/Transmitter (HSUART) serial ports. Each UART port signal may be shared and MUXed with other functional blocks (i.e., GPIO, SMBus, etc.).

**Figure 18-1. What is Covered in This Chapter**

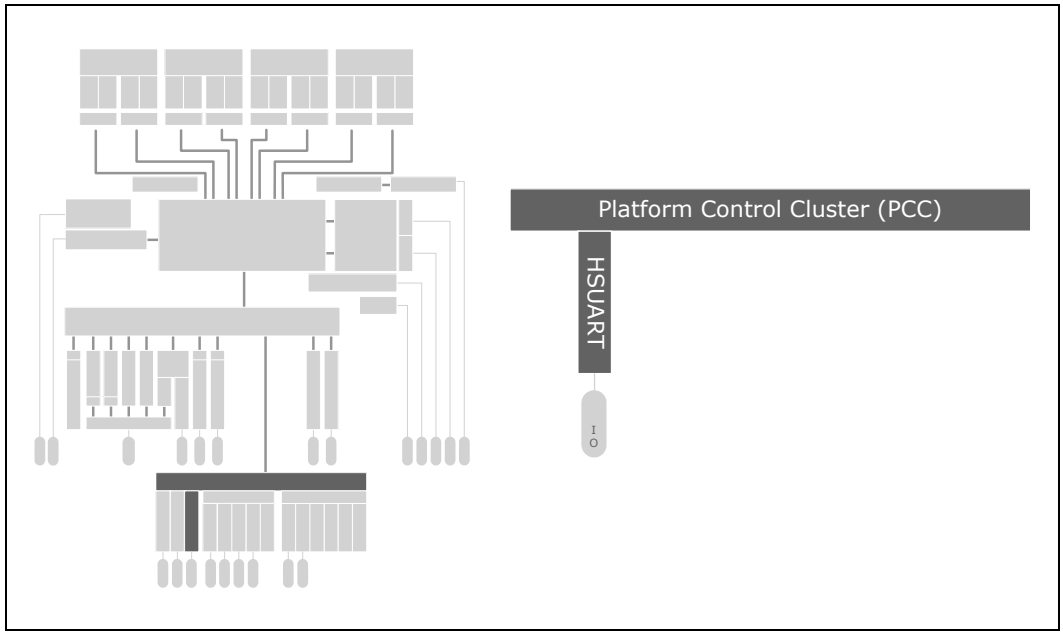



**Table 18-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| PC16550D UART Specification | Rev C | May 2015 | *PC16550D Universal Asynchronous Receiver/Transmitter With FIFOs (Rev. C)* |

1. At most, three of the SoC four High Speed Universal Asynchronous Receiver/Transmitter (HSUART) controllers can be active. One set of the SoC HSUART pins is configurable to be either the Host Root HSUART or else the Innovation Engine (IE) HSUART.

## 18.1 Signal Descriptions

The signal descriptions are shown in Table 18-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction/Type column of Table 18-2 is interpreted as follows:

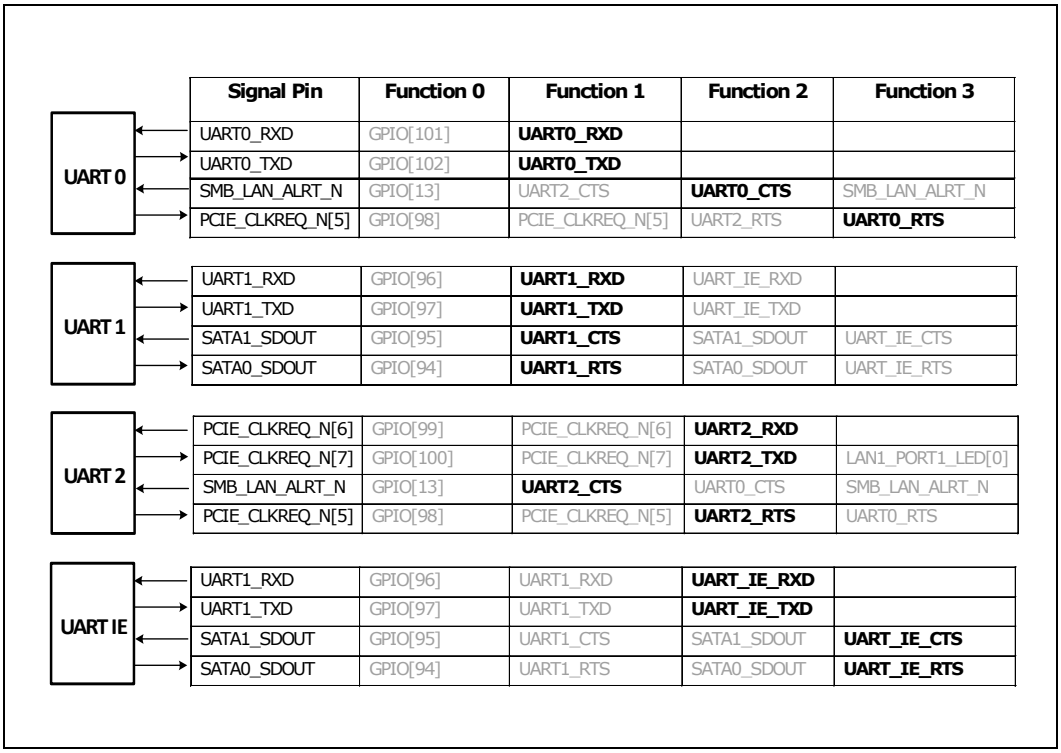
- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.

### Table 18-2. Signal Names and Descriptions

| Signal Names | Direction | Description |
|---|---|---|
| **UART 0** | | |
| UART0_RXD | I | **UART0 Receive Data:** Active-high input |
| UART0_TXD | O | **UART0 Transmit Data:** Active-high output |
| UART0_CTS | I | **UART0 Clear to Send:** Active-high input |
| UART0_RTS | O | **UART0 Request to Send:** Active-high output |
| **UART 1** | | |
| UART1_RXD | I | **UART1 Receive Data:** Active-high input |
| UART1_TXD | O | **UART1 Transmit Data:** Active-high output |
| UART1_CTS | I | **UART1 Clear to Send:** Active-high input |
| UART1_RTS | O | **UART1 Request to Send:** Active-high output |
| **UART 2** | | |
| UART2_RXD | I | **UART2 Receive Data:** Active-high input |
| UART2_TXD | O | **UART2 Transmit Data:** Active-high output |
| UART2_CTS | I | **UART2 Clear to Send:** Active-high input |
| UART2_RTS | O | **UART2 Request to Send:** Active-high output |
| **UART Innovation Engine**[1] | | |
| UART_IE_RXD | I | **UART_IE Receive Data:** Active-high input |
| UART_IE_TXD | O | **UART_IE Transmit Data:** Active-high output |
| UART_IE_CTS | I | **UART_IE Clear to Send:** Active-high input |
| UART_IE_RTS | O | **UART_IE Request to Send:** Active-high output |

1. All Innovation Engine (IE) related material described in this section can be reference to Chapter 37, "Innovation Engine".

## Figure 18-2. UART MUXed Signals



## 18.2 Feature List

### 18.2.1 UART Features

- 16550 and 16750 compliant UART controllers
- 64-byte FIFO for each receiver and transmitter
- Programmable interrupt trigger levels for FIFOs
- Baud generation based on programmable multipliers and divisors
- Supports programmable baud rates from 300 bps to 3.6864 Mbps
- Data rate can be programmed using the multiplier and divisor of the direct digital synthesizer.
- Auto-baud rate detection capability
- Flow control by hardware (HW) or software (SW)
- Configurable data format
- Data bits: 5, 6, 7, or 8 bits
- Even, odd, or no parity generation
- 1, 1.5, 2 stop bits
- False start-bit detection
- Line break generation/detection
- Prioritized interrupt
- Supports DMA and non-DMA mode for data access
- Complete status report capabilities

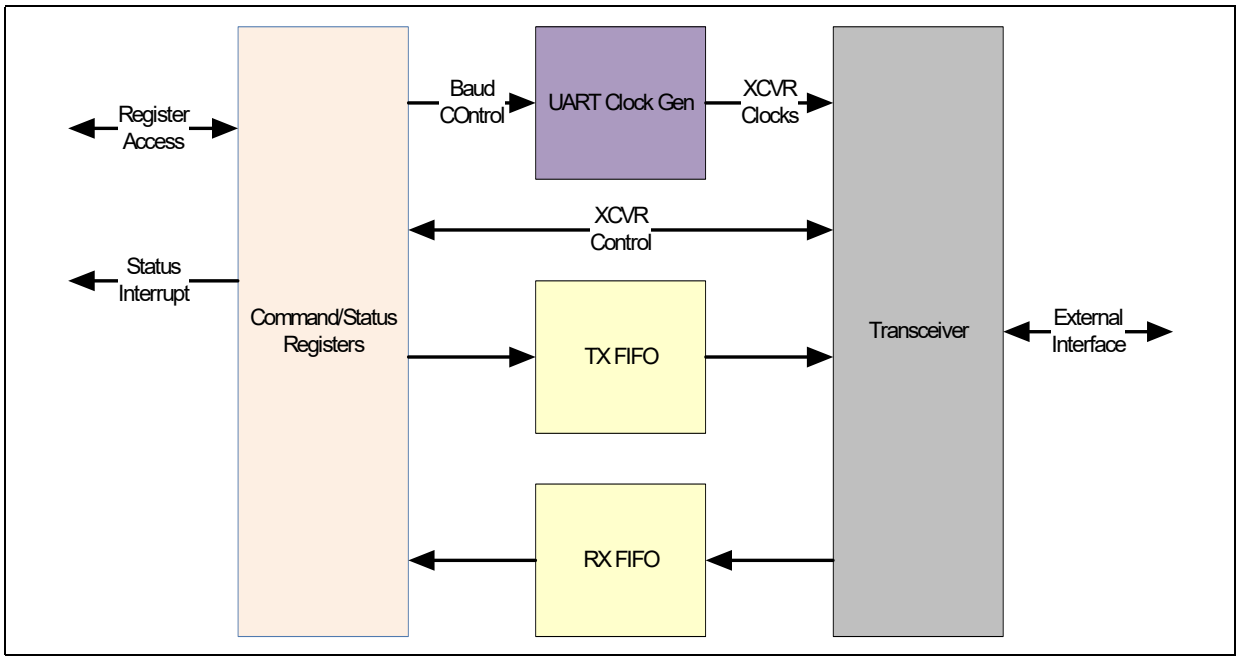
### 18.2.2 DMA Features

- Supports 32-bit data and addressing
- Dedicated TX and RX channel for each UART
- Each DMA channel has four descriptors
- Configurable interrupt generation for DMA operation
- Support for timeout
- Supports a single outstanding request at a time
- Uses round robin arbitration among active channels
- Support to generate non-DW aligned memory requests

## 18.3 Architectural Overview

The UART operates per the 16550 component specifications with optional DMA support. The UART can operate in legacy or enhanced mode. The first eight bytes of the 16550 register are set in addition to other registers for the enhanced UART mode.

**Figure 18-3. UART Core Block Diagram Overview**



### 18.3.1 UART Hiding and Locking

The 3 IA Host space UART devices are implemented as PCI devices that are discovered by PCI enumeration. The system BIOS always preforms PCI enumeration and the OS may re-enumerate the bus. This can cause the UART devices to move between BIOS and OS.

If your system does not use any or all of the UARTs the BIOS can hide them before preforming the PCI enumeration. The BIOS does this by writing to each UART PCI configuration space register FUNC_RDCFG_HIDE. UART devices must be hidden from UART 3, then UART 2, and lastly UART 1.

If your system requires the UART to be at a legacy COM 1, 2, 3 or 4 address then before PCI enumeration BIOS can assign the legacy I/O address and MMIO address space (if DMA is required). BIOS must also enabled the I/O address decoding by writing bit 0 of CMD_STAT_REG to 1'b1 and if used the MMIO address decoding by writing bit 1 to 1'b1. Then BIOS would hide the UART via PCI configuration space register FUNC_RDCFG_HIDE. In addition BIOS must setup the SoC to route the COM address to the UART and not to the LPC interface. See Table 3-3, “Host Fixed I/O Address Space” on page 176 for the COM address assigned on what additional registers need to be programmed.

If any of the UART devices are left as PCI enumerable (not hidden or put at a legacy COM address) they must be from low to high UART 1, then UART 2 then UART3.

**Table 18-3. Legacy COM I/O Ports**

| COM Port | I/O Ports |
|---|---|
| COM1 | 3F8h – 3FFh |
| COM2 | 2F8h – 2FFh |
| COM3 | 3E8h – 3EFh |
| COM4 | 2E8h – 2EFh |

### 18.3.2 Legacy Mode

In the legacy mode, the UART operates per the 16550 component specifications. The UART requests the eight bytes of I/O and 256 bytes of memory space using the corresponding PCI BAR0 (I/O Base Address Register, IOBA) and BAR1 (Memory Base Address Register, MEMBA). MEMBA is only required if your system uses the DMA feature.

The UART TX and RX FIFO depth can be configured to 64 bytes as opposed to only 16 bytes supported by 16550. This UART does not support DMA in legacy mode and does not comply with the DMA signaling mechanism outlined in the 16550 specifications.

### 18.3.3 Enhanced Mode

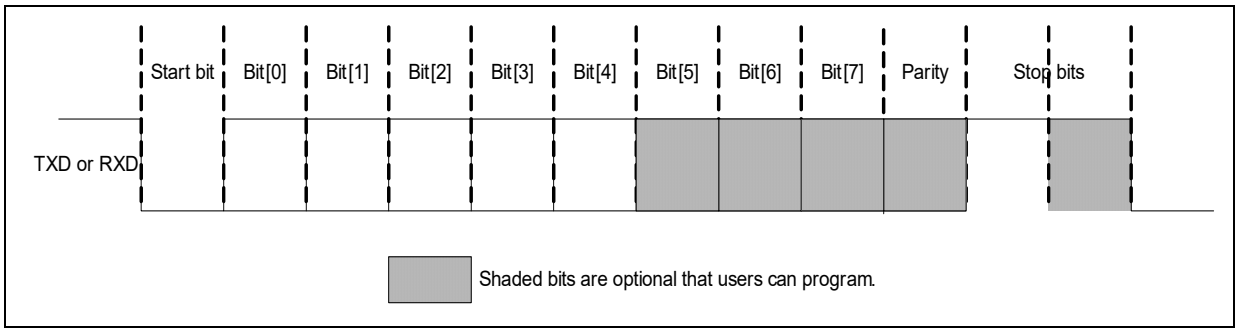
In the enhanced mode, the enhanced register set is available in addition to the legacy mode Command/Status Registers. The UART requests 256 bytes of the memory space per function using BAR1. In the enhanced mode, the memory space is required to be enabled. The UART supports the following additional features in the enhanced mode:

- Programmable UART base frequency using a clock multiplier and divisor
- FIFO size up to 64 bytes
- Baud rates higher than 115.2 kbps (see Table 18-4)
- Auto-baud detection capability
- Auto-flow mode
- DMA capability

## 18.3.4 UART Function

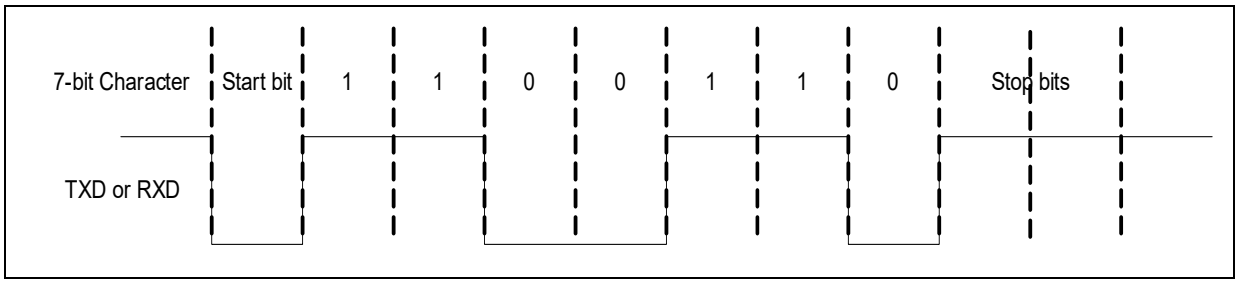
The UART transmits and receives data in bit frames. Each data frame is between 7 and 12 bits long, depending on the size of data programmed, whether parity is enabled or disabled, and the number of stop bits. The frame begins with a start bit that is represented by a high-to-low transition. Next, 5 to 8 bits of data are transmitted, beginning with the least significant bit. An optional parity bit follows, if enabled. The data frame ends with 1, 1.5, or 2 stop bits (as programmed), represented by 1 or 2 successive bit periods of a logic one.

**Figure 18-4. UART Data Transfer Flow**




The UART serial bit frames are encoded, where a 1 is represented by a line transition, and a 0 is represented by no line transition. Figure 18-5 shows the coded full 7-bit character 1100110 with no parity and 2 stop bits. The LSB of a byte is transmitted first.

**Figure 18-5. 7-bit Character Encoding**




Out of reset, the UART transceiver is enabled with the default register configuration. The UART transceiver can be disabled by clearing the UART Unit Enable (UUE) bit of the Auto-Baud Control register. When UUE is set, the receiver looks for the start bit of a frame and the transmitter sends data if data is available in the Transmit FIFO or Transmit Holding register. Disabling the transceiver with the UUE bit does not clear the data in the receive/transmit FIFO. Re-enabling the UART will continue transmission/reception with the original data.

Each UART has a transmit FIFO and a receive FIFO and each can store 64 characters of data. There are two separate methods for moving data into and out of the FIFOs: interrupts and polling. The UART also supports the full and half-auto flow control mode for transmission and receive. The UART baud rate can be configured or it also can support the auto-baud rate detection feature. Key UART modes and features are discussed in subsequent sections.

## 18.3.5 FIFO Interrupt Mode Operation

### 18.3.5.1 Receive Interrupt

When the receive FIFO (FCR [TRFIFOE] = 1) and the receiver interrupt (IER [RDA] = 1) are enabled, the following receiver interrupts can occur:

- The receive data available interrupt is asserted when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
- The IIR [IID] receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
- The Data Ready bit (LSR [DR]) is set to 1 as soon as a character is transferred from the Shift register to the receive FIFO. This bit is reset to 0 when the FIFO is empty.

### 18.3.5.2 Timeout Interrupt

When the receive FIFO and receiver interrupt are enabled, a character timeout interrupt (IIR [TOD]) will occur to signal the presence of trailing bytes. The interrupt is cleared and the timer is reset when a character is read from the receiver FIFO. If a timeout interrupt has not occurred, then the timeout timer is reset after a new character is received or after the processor reads the receiver FIFO. Character timeout will occur when all the following conditions are true:

- At least one character is in the FIFO.
- A character has not been received for the amount of time it takes to receive four or more characters at the current baud rate.
- The FIFO has not been read for the amount of time it takes to receive four or more characters at the current baud rate.

### 18.3.5.3 Transmit Interrupt

A character has not been received for the amount of time it takes to receive four or more characters at the current baud rate.

The FIFO has not been read for the amount of time it takes to receive four or more characters at the current baud rate.

When the transmit FIFO (FCR [TRFIFOE] = 1) and the transmitter interrupt (IER [TDRQ] = 1) are enabled, the following receiver interrupts can occur:

- When FCR [TIL] = 0, the transmit data request interrupt occurs whenever the transmit FIFO is empty. The interrupt is cleared as soon as the Transmit Holding register is written or the IIR is read.
- When FCR [TIL] = 1, the transmit data request interrupt occurs when the transmit FIFO is empty or the FIFO level reaches the half empty mark from more than half full. The interrupt is cleared as soon as the Transmit Holding register is written or the IIR is read.

Users could cause the UART transmit FIFO to overflow if too many characters are written. FIFO underflow does not cause an error as the UART will wait for the transmit FIFO to be serviced.

### 18.3.6 FIFO Polled Mode Operation

With the FIFOs enabled (FCR [TRFIFOE] = 1), clearing IER puts the serial port in the FIFO polled operation mode. Because the receiver and the transmitter are controlled separately, either one or both can be in the polled operation mode. In this mode, software checks the receiver and transmitter status via the Line Status Register (LSR). The processor polls the following bits for receive and transmit data service.

**Receive Data Service**

Poll the data ready (LSR [DR]) bit that is set when 1 or more bytes remain in the receive FIFO or Receive Buffer Register (RBR).

**Transmit Data Service**

Poll the transmit data request LSR [TDRQ] bit that is set when the transmit FIFO or Transmit Holding Register (THR) is empty.

Poll the transmitter empty LSR [TEMT] bit that is set when the transmitter is empty.

### 18.3.7 Auto-flow Control Operation

Auto-flow control uses the Clear-to-Send (CTS) and Request-to-Send (RTS) signals to automatically control the flow of data between the UART and the external serial device. When auto-flow is enabled, the remote device is not allowed to send data unless the UART asserts RTS low. If the UART de-asserts RTS while the remote device is sending data, the remote device is allowed to send one additional byte after RTS is de-asserted. An overflow could occur if the remote device violates this rule. Likewise, the UART is not allowed to transmit data unless the remote device asserts CTS low. This feature increases system efficiency and eliminates the possibility of a receive FIFO overflow error due to long interrupt latency.

The auto-flow mode can be used in two ways: full auto-flow, automating both CTS and RTS; and half auto-flow, automating only CTS. Full auto-flow is enabled by writing a 1 to bits Modem Control Register (MCR) - MCR [AFE][1] and MCR [RTS] of the MCR. Auto-CTS-only mode is enabled by writing a 1 to MCR [AFE] and a 0 to MCR [RTS] of the MCR register.

The auto-flow mode can only be used in conjunction with the FIFO mode. During the auto flow mode, the modem interrupt should not be enabled.

The following are descriptions of RTS and CTS in both modes.

**RTS (UART Output)**

In full auto-flow mode, RTS is asserted when the UART FIFO is ready to receive data from the remote transmitter. This occurs when the amount of data in the receive FIFO is below the programmable threshold value. When the amount of data in the receive FIFO reaches the programmable threshold, RTS is de-asserted. It will be asserted once again when enough bytes are removed from the FIFO to lower the data level below the threshold.

**CTS (UART Input)**

In full or half auto-flow mode, CTS is asserted by the remote receiver when the receiver is ready to receive data from the UART. The UART transmitter checks CTS before reading the next byte from the FIFO and will not read the byte until CTS is low. If CTS goes high while the transfer of a byte is in progress or about to transfer a byte, the transmitter will complete the transmission of this byte.

1. Auto Flow Enable (AFE).

### 18.3.8 Loopback Operation

The loopback mode is on when MCR [LOOP] is set. During the loopback mode, the following will occur:

- The UART serial out TXD is set to an idle state. The UART serial input RXD is internally disconnected. The internal TX line is looped back into the internal RX line.
- The modem inputs are internally disconnected and modem outputs are connected to the disconnected inputs. The modem outputs are set to the idle state.

In the loopback mode, data that is transmitted is immediately received. This feature allows checking of the transmit and receive data paths of the UART.

In the loopback mode, receive and transmit interrupts are fully functional if enabled. The modem interrupts are also functional if enabled, but the sources of interrupt are the lower 4 bits of the MCR.

*Note:* Full auto-flow and loopback mode can be enabled simultaneously for efficient diagnostic testing. Half auto-flow and loopback mode should not be enabled simultaneously.

### 18.3.9 Auto-Baud Detection

Each UART supports auto-baud rate detection. This feature is only available in enhanced mode. When enabled, the UART counts the number of reference clock cycles within the receive start-bit pulse. The frequency of the reference clock is set when the UART clock gen is initialized. The default UART clock is 1.8432 MHz. The number of UART clock pulses in the start bit is written into the Auto-Baud Count Register (ACR) and is used to determine the baud rate. When the ACR is written, an auto-baud lock interrupt is generated (if enabled), and the UART can automatically program the Divisor Latch registers with the appropriate baud rate. If preferred, the processor can read the Auto-Baud Count register and use this information to program the Divisor Latch registers with a baud rate calculated by the processor. After the baud rate has been programmed, the host is responsible to verify that the predetermined characters (usually AT or at) are being received correctly.

If the UART is allowed to program the Divisor Latch registers in the auto-baud mode, users can choose between two methods for auto-baud calculation: table-based and formula-based. The method is selected via bit Auto-baud Table (ABT) of the Auto-Baud Control Register (ABR). When the formula-based method is used, the baud rate, based on the number of UART clock pulses in the start bit, can be programmed by the UART. This method works well for higher baud rates, but could possibly fail below 28.8 Kbps because of the fine resolution of the clock measuring the duration of the start bit, especially when the duration of the start bit is not exactly equal to the nominal due to the use of slow level translators. The actual baud rate may differ by more than one percent of its target.

Once the baud rate is detected the first time, the auto-baud circuitry will disarm itself by clearing bit ABE of the Auto-Baud Control Register (ABR). The Auto-Baud Enable (ABE) bit must be written as 1 by software to rearm the circuitry.

For the auto-baud rate detection circuit to work correctly, the first data bit transmitted after the start bit must be logic 1. If logic 0 is transmitted instead, the auto-baud circuit will count the 0 as part of the start bit, resulting in an incorrect baud rate being programmed into the Divisor Latch Low (DLL) and Divisor Latch High (DLH registers. The input signal with a baud rate less than 150 bps will result in undefined behavior of the auto-baud detection circuitry.

### 18.3.10 UART Clocking

The UART clock is derived from internal SoC clock system.

FUART = 44236800 Hz

### 18.3.11 UART Baud Rate Generation

The baud rates are generated from the UART base frequency (FUART) by programming the Divisor Latch Access Bit (DLAB) from Line Control Register (LCR.DLAB) and Pre-Scalar (PSR) registers.

FBAUD = FUART / (PSR/ DLAB)

To get the supported baud rates as mentioned in table below, use FUART = 44236800 Hz.

**Table 18-4. Supported Baud Rates**

| FUART (Hz) | PSR | LCR.DLAB | FBAUD (bps) |
|---|---|---|---|
| 44236800 | 0C | 0001h | 3686400 |
| 44236800 | 10C | 0003h | 921600 |
| 44236800 | 10C | 0006h | 460800 |
| 44236800 | 10C | 0009h | 307200 |
| 44236800 | 10C | 000Ch | 230400 |
| 44236800 | 10C | 000Fh | 184320 |
| 44236800 | 10C | 0012h | 153600 |
| 44236800 | 10C | 0018h | 115200 |
| 44236800 | 10C | 0030h | 57600 |
| 44236800 | 10C | 0048h | 38400 |
| 44236800 | 10C | 0090h | 19200 |
| 44236800 | 10C | 0120h | 9600 |
| 44236800 | 10C | 0180h | 7200 |
| 44236800 | 10C | 0240h | 4800 |
| 44236800 | 10C | 0300h | 3600 |
| 44236800 | 10C | 0480h | 2400 |
| 44236800 | 10C | 0600h | 1800 |
| 44236800 | 10C | 0900h | 1200 |
| 44236800 | 10C | 1200h | 600 |
| 44236800 | 10C | 2400h | 300 |

## 18.3.12 I/O and Memory Base Access Space

The UART supports 8 bytes of I/O space and 256 bytes of memory space to access registers of the UART. The lower 8 bytes are compliant to the 16550 UART. While addresses beyond 8 bytes give access to more advance features which include an auto-baud lock mechanism, DMA channel access, and higher baud rates.

Configure BAR0 (I/O Base Address Register, IOBA) by placing the device anywhere in the 16-bit I/O address space.

Configure BAR1 (Memory Base Address Register, MEMBA) by placing the device anywhere in the 32-bit memory address space.

## 18.3.13 Interrupt

The UART supports MSI messages to communicate interrupts.

In the event the MSIMC[MSIE] is set to 1, it is mandatory to configure the MSI address and MSI data properly in order to identify the MSI interrupt from this device.

**Table 18-5. Interrupt Enabling**

| | CMD[ID] = 0 | CMD[ID] = 1 |
|---|---|---|
| MSIMC[MSIE] = 1 | MSI sent via primary Interface | No interrupt to host |

## 18.3.14 Memory Address MAP

The memory map is split between the UART Memory Mapped registers Table 18-10 and the DMA channel registers Table 18-11. The UART registers are mapped to the first 128 bytes in the memory map. Each DMA channel takes the next 64 bytes. The total memory requirement for a single function is shown in Table 18-6.

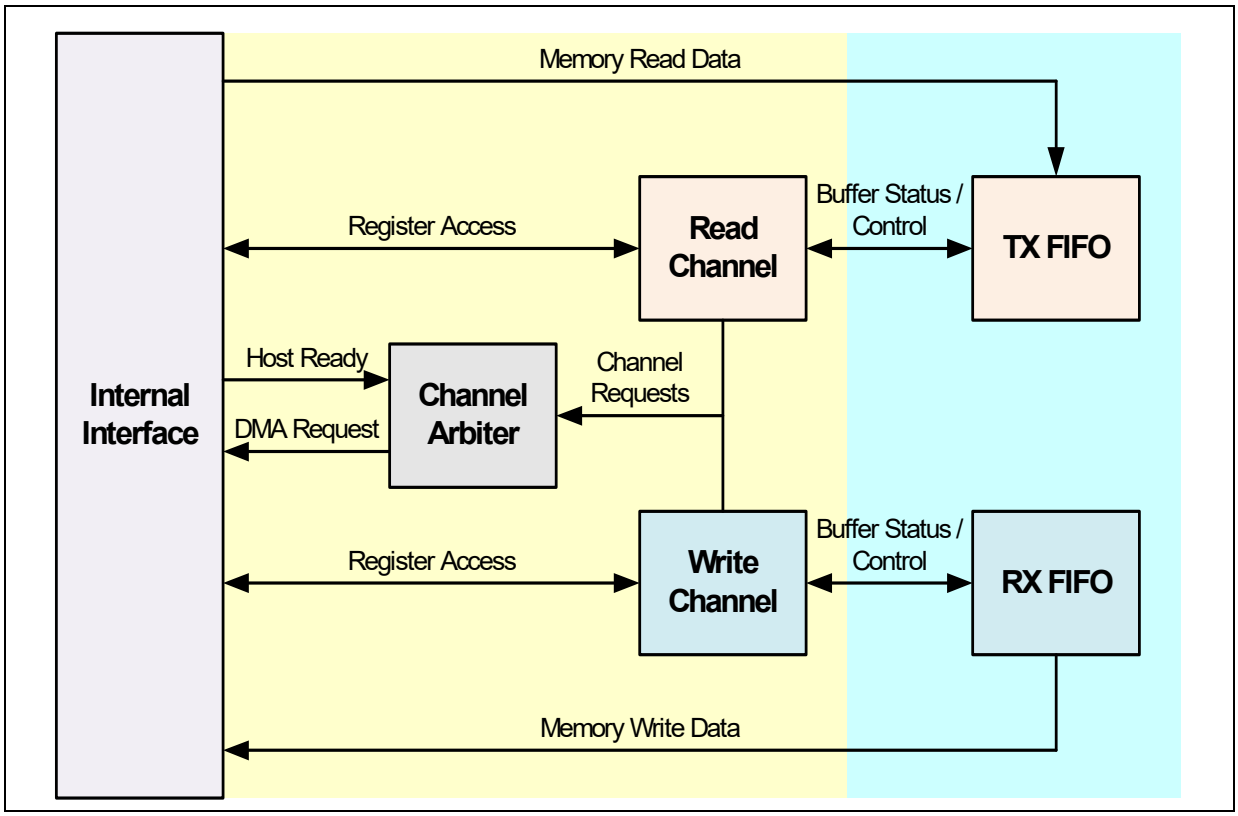
**Table 18-6. UART Memory Space**

| Block | Memory Size | Base Address | Description |
|---|---|---|---|
| UART[n] | 128 bytes per function | 0x00 | UART memory space requirement |
| DMA Channel[0] | 64 bytes for read channel | 0x80 | DMA memory space read requirement |
| DMA Channel[1] | 64 bytes for write channel | 0xC0 | DMA memory space write requirement |

### 18.3.15 DMA

The UART provides optional support for direct memory access transactions that isolate the CPU from the need to deliver large chunks of data to or from the main memory to the UART FIFOs.

**Figure 18-6. DMA Block Diagram**




Each UART has a dedicated read and write channel that will handle read and writes to the host memory. DMA channels have around robin arbiter that arbitrates the requests from all the channels.

Each channel has four descriptor sets. Each has its own Start Address Register (SAR) and Transfer Size Register (TSR). The channel logic chases each descriptor on a round robin manner, i.e., once descriptor 0 is completely done, it moves to descriptor 1, and once descriptor 3 is done it goes back to descriptor 0.

Each descriptor that is active and selected for chasing, delivers the start address and the transfer size. The channel logic decides the actual transfer size, based on the availability of data/space in the buffer and based on the minimal/maximal transfer size programmed to that channel. The descriptor is described as following:

- Start address + actual transfer size -> start address
- Transfer size - actual transfer size -> transfer size
- The design ensures that actual transfer size <= transfer size.

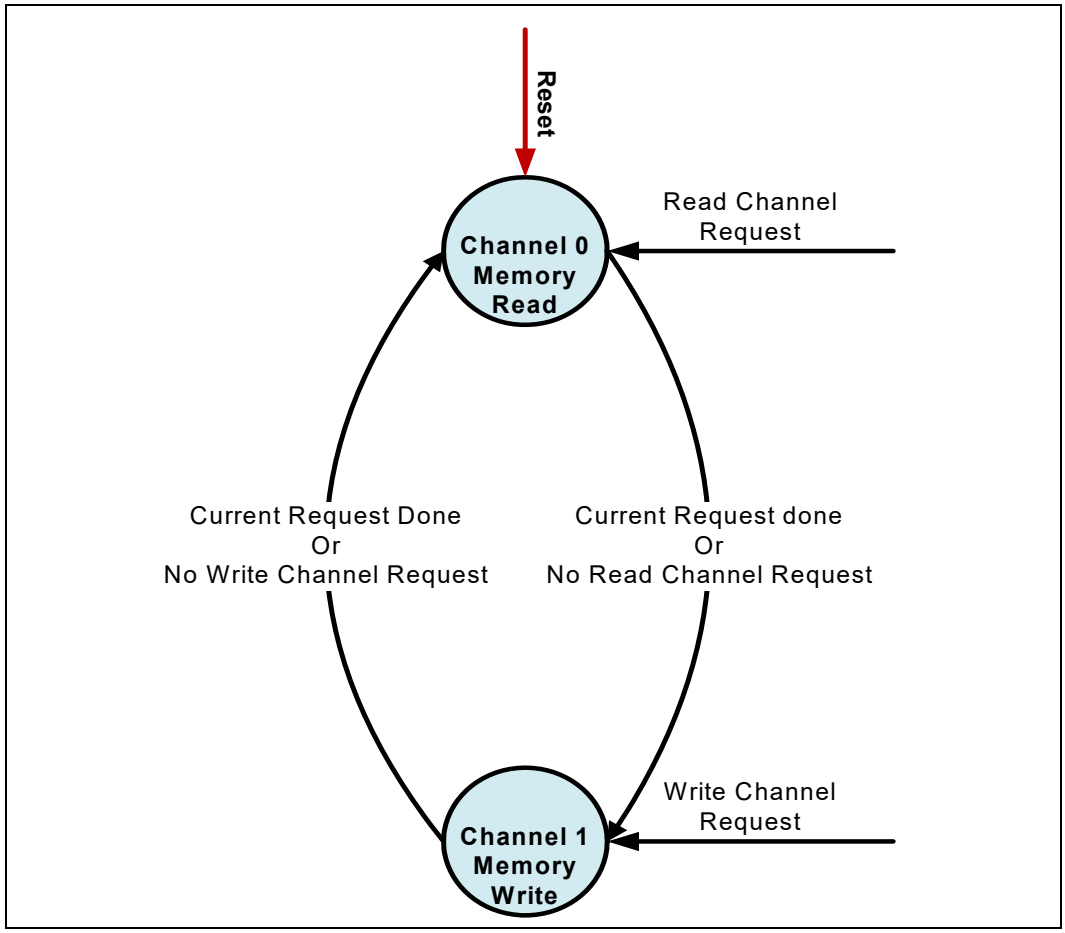
Once the descriptor transfer size reaches 0, the descriptor is considered to be completed and the channel logic can move to chasing the next descriptor.

### 18.3.15.1 Channel Arbitration

DMA channel arbitration has the following key functionality:

- Round robin arbitration scheme is used.
- The arbiter scans all the channels and whenever a channel has a valid request, the arbiter will send the request to the host interface if it is ready.
- Once the current request is done, the arbiter moves to the next channel, even if the current channel wishes to perform another transfer.

**Figure 18-7. Logical Representation of Channel Arbitration**



### 18.3.15.2 Descriptor Arbitration

The descriptor arbitration within each DMA channel has following key functionality:

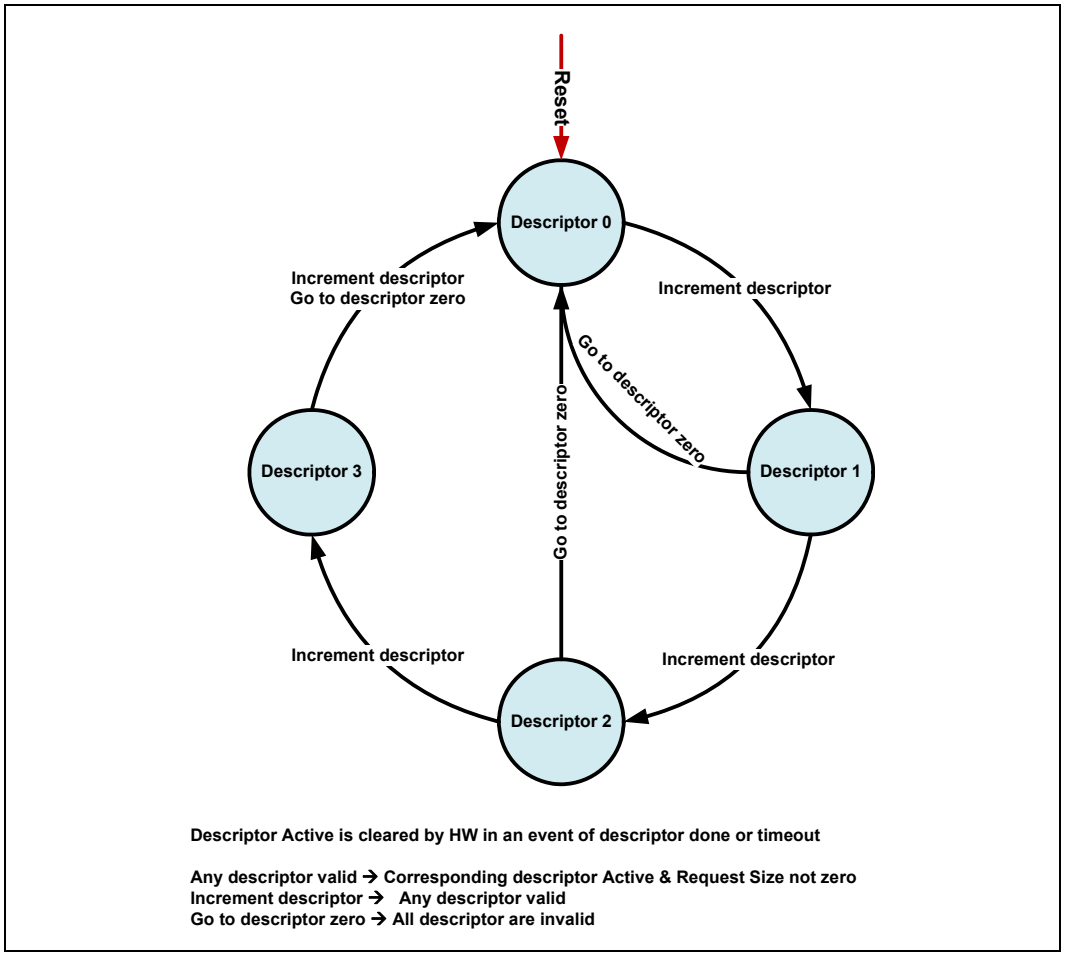
- Round Robin progresses 0 -> 1 -> 2 -> 3 -> 0 and so on.

On reset, the current descriptor points to descriptor 0. Descriptor arbitration is a function of the descriptor active, descriptor request size, and descriptor timeout for the corresponding descriptor. The descriptor select logic advances from one descriptor to the next once the transfer size on the current selected descriptor is down to 0 or a timeout condition is encountered during these events, the descriptor active bit will be cleared which results in an invalid descriptor.

When all descriptors are invalid, the current descriptor will immediately point to descriptor 0. The descriptor select logic starts with the first programmed valid descriptor. The current descriptor status is logged into the Channel Status register.

Any descriptor with a non-zero request size but an inactive descriptor active bit will not place the request to the channel arbiter.

**Figure 18-8. Logical Representation of Descriptor Arbitration**



### 18.3.16 MSI Generation

The UART has the capability to generate an MSI whenever the interrupt event happens within the UART or DMA units. The host needs to program the PCI MSI configuration registers to enable the MSI generation. The host needs to read the Capabilities Pointer (CP) PCI configuration register which points to the MSI Capability ID (MSICID) register. Then, the host needs to program the MSI Message Control (MSIMC) register, MSI Message Address (MSIMA) register, and MSI Message DATA (MSIMD) register to enable MSI generation for the UART. An MSI will be sent upstream as a posted write request to the address in MSIMA with the data value in MSIMD.

Once an MSI is enabled, legacy interrupt over sideband will be disabled. A single MSI is sent upstream whenever any of the interrupt is asserted. In the event of MSI software needs to read the relevant registers from all the three interrupt sources: UART, DMA read channel, and DMA write channel.

## 18.4 UART Programming Recommendation

This section is provided as recommendation steps for configuration sequencing for the UART programming.

### 18.4.1 Setup Access Space

The UART supports 8 bytes of the I/O space and 256 bytes of the MEM space to access registers of the UART. The lower 8 bytes are compliant to the 16550 UART. While addresses beyond 8 bytes give access to more advances features which includes the auto baud lock mechanism, DMA channel access, and higher baud rates.

Configure BAR0 (I/O Base Address Register, IOBA) by placing the device anywhere in the 16-bit I/O address space.

Configure BAR1 (Memory Base Address Register, MEMBA) by placing the device anywhere in the 32-bit memory address space.

### 18.4.2 Setup Interrupt

The UART supports MSI messages to communicate interrupts.

In the event that MSIMC[MSIE] is set to 1, it is mandatory to configure the MSI address and MSI data properly in order to identify an MSI interrupt from this device.

**Table 18-7. Interrupt Enabling**

| | CMD[ID] = 0 | CMD[ID] = 1 |
|---|---|---|
| MSIMC[MSIE] = 1 | MSI sent via primary interface | No interrupt to host |

### 18.4.3 Space Enable

Once configuration of the PCI space is done, the host should enable the required space (Memory or I/O). Also, the host should allow the device to initiate a transaction as a master.

To enable memory space CMD[MSE] =1'b1 (1st bit, binary, set to 1)

To enable I/O space CMD[IOSE] = 1'b1

To enable device as a master CMD[BME] = 1'b1

### 18.4.4 DMA Programming

Even though the DMA channel number is fixed (i.e., Channel 0 is for DMA Rx and Channel 1 is for DMA Tx), it is mandatory to program CHnCR with the appropriate channel direction value. Note that "CHn" from register name represented to DMA Channel 0 and DMA Channel 1 for this chapter.

The UART has FIFO that holds 64 bytes in each direction (UART Tx and UART Rx). the same number must be configured to the Buffer Size register Channel n=channel number Status Register (CHnBSR).

The host has to enable the FIFO mode of the UART when needed to operate with DMA (FCR[TRIFIFOE]).

### 18.4.5 DMA Descriptor Use/Reuse

In the case when the DMA is enabled, every DMA channel supports four descriptors. Descriptor evaluation happens sequentially from 0 to 3 and then rolls over to 0 again.

It is important to know which descriptor to be evaluated first. Intel recommends to program the descriptor in a sequence rather than randomly.

Once the DMA indicates it is done with the transfer, the host must program the following registers in order for the DMA engine to chase the descriptors again. Failure in the sequence of this program may result in the DMA engine not working when the same descriptor is used again.

1. Deactivate the channel before modifying any descriptor values Channel n=channel number Control Register (CHnCR).
2. Disable all the descriptors (CHnDCR).
3. Program each descriptors of the respective channel with the appropriate value of SAR and TSR (CHnD0/1/2/3SAR, CHnD0/1/2/3TSR).
4. Change the minimum transfer size if required (CHnMOTSR).
5. Activate the channel (CHnCR).
6. Enable the required descriptors with the appropriate interrupt enable bits (CHnDCR).

### 18.4.6 Programming Optimization

#### 18.4.6.1 Optimization Descriptor Programming Activities During DMA

Program TSR of each descriptor with a sufficient high value.

In the case of the DMA Rx, the host knows how many characters are required to be transferred. Therefore, it should use the full capacity of the TSR and SAR combination to transfer enough data at the UART Tx signals.

In the case of the DMA Tx, the host does not have the knowledge how much data is going to be received, but still the host can enable the timeout interrupt CHnDCR[27:24] for each descriptor. Also, program the fairly large value in TSR to minimize the descriptor programming cycles.

*Note:* The descriptors are still active when the DMA timeout happens, but the DMA done is not set. The DMA timeout acts as just a notification to the host that nothing is being received in last four character cycles. Also, the DMA engine empties the buffer towards the host with the appropriate address and length.

#### 18.4.6.2 Optimization Access Time By UART During DMA

Channel minimum transfer size (CHnMTSR) instructs the DMA engine when to initiate a request towards the internal interface to the read/write data from the host memory. That means a smaller value shall initiate more requests with less payload size, and a higher value shall initiate less requests with a higher payload size.

This value must not increase the buffer size value in order to prevent overflow (in the case of DMA Tx) or deadlock (in the case of DMA Rx). For SoC I/O fabric UART, the maximum value that can be programmed is 64, and the minimum value that can be programmed is 1.

Intel recommends to program this register with value = 60 on the safe side and to optimize the Dword boundary access.

### 18.4.6.3 Optimization Interrupt Handling

The UART has two sources of interrupt and shares a common interrupt notification source to communicate an interrupt to the host. If an interrupting condition from the source is not cleared, a subsequent interrupt shall not be communicated. So it is very important to clear the interrupt source upon receipt by the host.

#### 18.4.6.3.1 DMA Interrupt Handling

This section contains interrupt sources which can occur only when the UART is running in the DMA mode.

**Descriptor Done Interrupt**

When the DMA engine transfers the configured amount of characters (configured via TSR), the DMA engine asserts this interrupt to notify the host to reprogram the respective descriptor. In order to avoid more interrupts from this source, the host may chose to enable this interrupt only for the upper descriptor. For example, if three descriptors are programmed, counting from 0 -> 1 ->2, enable a done interrupt only for descriptor 2. This will skip an interrupt to the host when descriptor 0 and descriptor 1 are completed.

The cost that the host needs to pay is to keep track of which descriptors were programmed before.

This interrupt gets cleared upon reading the DMA interrupt status register.

**Descriptor Timeout Interrupt**

This interrupt acts as more of a notification of any action for the host. A timeout interrupt can never happen for the DMA Rx channel (Channel 0). Upon character timeout, the HSUART interrupts the host and then activates DMA Tx to flush the trailing bytes from the buffer.

This interrupt gets cleared upon reading the DMA interrupt status register.

#### 18.4.6.3.2 UART Interrupt Handling

This section contains interrupt sources from the UART. The 16550 has the following five types of interrupts.

**Receive Data Available Interrupt**

This interrupt notifies the host that at least a configured number of characters are present in the buffer. The lower the number, the more frequent interrupts can be expected.

The threshold for this interrupt can be controlled through the FCR[ITL] value.

*Note:* In the DMA mode, this interrupt acts as a notification only. The DMA engine takes care of transferring the data to the host memory when the Rx buffer gets up to a minimum transfer size.

Programming this interrupt in the DMA mode may block any further interrupt communicated by the DMA or UART source if this interrupt is on. For instance, if FCR[ITL] is programmed to a value 00, upon every SoC I/O fabric transfer, which empties the Rx buffer, this interrupt gets de-asserted, and upon receiving the first character interrupt gets asserted.

Intel recommends to disable this interrupt when running with the DMA mode.

**Transmit Data Request Interrupt**

This interrupt notifies the host about availability for more characters in the transmit FIFO. The threshold for this interrupt can be controlled with the FCR[TIL] bit value.

*Note:* In the DMA mode, this interrupt acts as a notification only. The DMA engine takes care of transferring data from the host to the Tx buffer when a configured number of bytes are empty in the buffer.

Intel recommends to disable this interrupt while operating in the DMA mode.

**Receiver Line Status Interrupt**

This interrupt notifies the host that either error has been noticed in the UART reception or a break character is received. This interrupt notification is important for the host in order to receive error free characters.

Upon detection of line status interrupt, further decoding can be done by reading LSR.

Intel recommends to enable this interrupt.

**Modem Interrupt**

This interrupt notifies the host about modem-related information and updates.

If running with a modem signal, Intel recommends to enable an interrupt.

### Receiver Timeout Interrupt

This interrupt only occurs in the FIFO mode.

If operating in the DMA mode, this interrupt acts as a notification only. The DMA engine takes care of transferring the trailing bytes to the host memory upon detection of a timeout.

If operating in the DMA mode, Intel recommends to disable this interrupt.

If operating in the FIFO mode but not in the DMA mode, Intel recommends to enable this interrupt.

### Additional Interrupt

This set of interrupts are introduced in addition to the 16550 recommended interrupts from the UART module.

### Auto Baud Lock Interrupt

When the auto baud lock feature (ABR[ABE]) and auto baud lock interrupt (ABR[ABLIE]) are enabled, upon getting locked on a receiving character, this interrupt gets generated.

This interrupt gets cleared when reading IIR.

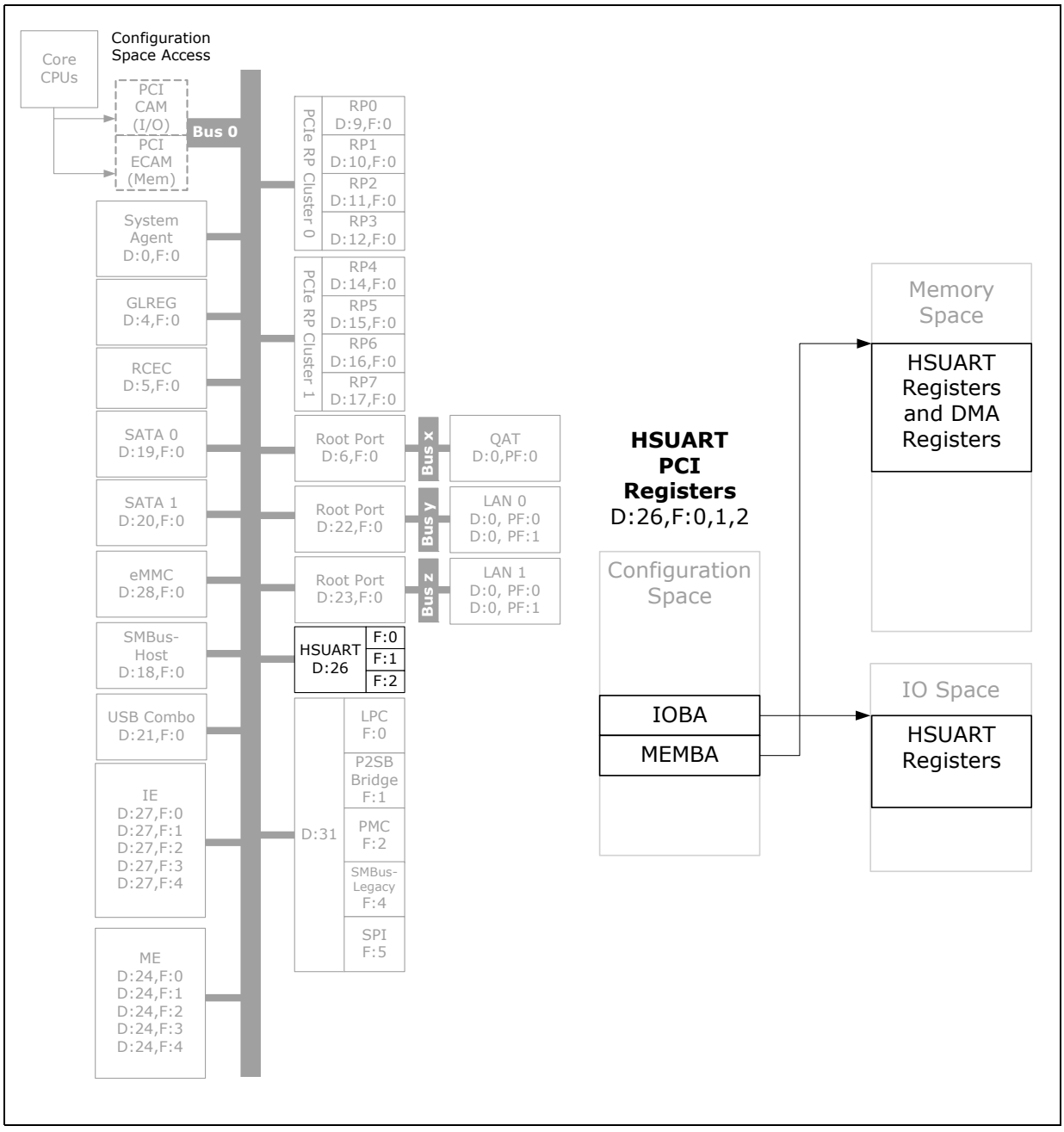
Intel recommends to enable this interrupt when operating with the auto baud lock feature. Also, Intel recommends not to transfer data unless the auto baud lock is achieved in the auto baud lock mode to ensure transfer integrity.

Enabling the auto baud lock interrupt when the auto baud lock feature is not enabled does not generate any interrupt to the host from the auto baud lock source.

## 18.5 Register Map

Figure 18-9 shows the associated registers from a system software viewpoint.

**Figure 18-9. Register Map**



### 18.5.1 PCI Configuration and Capabilities Registers

The UART PCI configuration registers:

- UART 0 = PCI Bus 0; Device 26 (decimal); Function 0
- UART 1 = PCI Bus 0; Device 26 (decimal); Function 1
- UART 2 = PCI Bus 0; Device 26 (decimal); Function 2

**Table 18-8. PCI Configuration Registers (HSUART_REGS_PCI)**

| Offset from Configuration Space of B0:D26:Fn (n=0,1,2) (Decimal) | Register Name | Description |
|---|---|---|
| 00h | VENDOR_DEVICE_ID | Vendor and Device ID Register |
| 04h | CMD_STAT_REG | Command and Status Register |
| 08h | REV_CLASS_CODE | Revision ID and Class Code Register |
| 09h | CACHE | Header Type Register |
| 10h | IOBA | I/O Base Address Register |
| 14h | MEMBA | Memory Base Address Register |
| 2Ch | SUBSYSTEM_VENDOR_ID | Subsystem Vendor ID Register |
| 34h | CAP_PTR | Capability Pointer Register |
| 3Ch | INTERRUPT | Interrupt Line Register |
| 40h | MSIC | MSI Capability Register |
| 44h | MSIA | MSI Message Address Register |
| 48h | MSID | MSI Message Data Register |
| 50h | PMC | Power Management Capability Register |
| 54h | PMCSR | Power Management Control and Status Register |
| 60h | DSC | Device Specific Control and Status Register |
| 74h | FUNC_RDCFG_HIDE | Function-Hide Register |

## 18.5.2 Memory Mapped Registers, BAR1 (Memory Base Address Register, MEMBA)

The memory map is split between the UART and DMA channel registers.

The UART registers are mapped to the first 128 bytes in the memory map. If the DMA is included in the design, each channel takes the next 64 bytes. The total memory requirement for a single function is shown in Table 18-9.

**Table 18-9. UART and DMA Memory Space**

| Block | Memory Size | Description |
|---|---|---|
| UART | 128 bytes per function | UART memory space requirement |
| DMA | 128 bytes per function | DMA memory space requirement<br>• 64-bytes for the read channel<br>• 64-bytes for the write channel |

### 18.5.2.1 UART Memory Mapped Registers

The UART is software compatible with 16650 and 16750 UART specifications. The first 8 bytes of the UART Command/Status Registers are a single byte register. These registers are DW wide.

*Note:* Software must not assume the existence of any Command/Status Registers which are not explicitly documented here.

**Table 18-10. UART Memory Mapped Registers**

| Offset from MEM Base Address MEMBA (FFFFFF00h) | LCR.DLAB | Access | Register Name | Description |
|---|---|---|---|---|
| 00h | 0 | RO | RBR | Receive Buffer Register |
| 00h | 0 | WO | THR | Transmit Holding Register |
| 00h | 1 | RW | DLL | Divisor Latch Low Byte Register |
| 01h | 1 | RW | DLH | Divisor Latch High Byte Register |
| 01h | 0 | RW | IER | Interrupt Enable Register |
| 02h | X | RO | IIR | Interrupt Identification Register |
| 02h | X | WO | FCR | FIFO Control Register |
| 03h | X | RW | LCR | Line Control Register |
| 04h | X | RW | MCR | Modem Control Register |
| 05h | X | RO/RC | LSR | Line Status Register |
| 06h | X | RO/RC | MSR | Modem Status Register |
| 07h | X | RW | SPR | Scratch Pad Register |
| 08h | X | RO | UART_FISR | UART Functional Interrupt Status Register |
| 20h | X | RO | RFOR | Receiver FIFO Occupancy Register |
| 24h | X | RW | ABR | Auto-baud Control Register |
| 28h | X | RO | ACR | Auto-baud Count Register |
| 30h | X | RW | PSR | Pre-Scalar Register |
| 34h | X | RW | UCMR | UART Clock Gen Multiplier Register |
| 38h | X | RW | UCDR | UART Clock Gen Divisor Register |

### 18.5.3 DMA Channel Registers

**Table 18-11. DMA Channel Memory Mapped Registers**

| Offset from MEM Base Address MEMBA (FFFFFF00h) | Access | Register Name | Description |
|---|---|---|---|
| 80h | RO | CH0SR | Channel Status Register |
| 84h | WO | CH0CR | Channel Control Register |
| 88h | WO | CH0DCR | Channel Descriptor Control Register |
| 90h | WO | CH0BSR | Channel Buffer Size Register |
| 94h | RO | CH0MTSR | Channel Minimum Transfer Size Register |
| A0h | WO | CH0D0SAR | Channel Descriptor 0 Start Address Register |
| A4h | WO | CH0D0TSR | Channel Descriptor 0 Transfer Size Register |
| A8h | WO | CH0D1SAR | Channel Descriptor 1 Start Address Register |
| ACh | WO | CH0D1TSR | Channel Descriptor 1 Transfer Size Register |
| B0h | WO | CH0D2SAR | Channel Descriptor 2 Start Address Register |
| B4h | WO | CH0D2TSR | Channel Descriptor 2 Transfer Size Register |
| B8h | WO | CH0D3SAR | Channel Descriptor 3 Start Address Register |
| BCh | WO | CH0D3TSR | Channel Descriptor 3 Transfer Size Register |
| C0h | RO | CH1SR | Channel Status Register |
| C4h | WO | CH1CR | Channel Control Register |
| C8h | WO | CH1DCR | Channel Descriptor Control Register |
| D0h | WO | CH1BSR | Channel Buffer Size Register |
| D4h | RO | CH1MTSR | Channel Minimum Transfer Size Register |
| E0h | WO | CH1D0SAR | Channel Descriptor 0 Start Address Register |
| E4h | WO | CH1D0TSR | Channel Descriptor 0 Transfer Size Register |
| E8h | WO | CH1D1SAR | Channel Descriptor 1 Start Address Register |
| ECh | WO | CH1D1TSR | Channel Descriptor 1 Transfer Size Register |
| F0h | WO | CH1D2SAR | Channel Descriptor 2 Start Address Register |
| F4h | WO | CH1D2TSR | Channel Descriptor 2 Transfer Size Register |
| F8h | WO | CH1D3SAR | Channel Descriptor 3 Start Address Register |
| FCh | WO | CH1D3TSR | Channel Descriptor 3 Transfer Size Register |

## 18.5.4 I/O Mapped Registers, BAR 0 (I/O Base Address Register, IOBA)

### 18.5.4.1 UART I/O Mapped Registers

The following UART registers are located in the I/O address space.

**Table 18-12. UART I/O Mapped Registers**

| Offset from I/O Base Address IOBA (FFFFFFF8h) | LCR.DLAB | Access | Register Name | Description |
|---|---|---|---|---|
| 00h | 0 | RO | RBR_THR_DLL | Receive Buffer Register (RBR) |
| 00h | 0 | WO | RBR_THR_DLL | Transmit Holding Register (THR) |
| 00h | 1 | RW | RBR_THR_DLL | Divisor Latch Low Byte Register (DLL) |
| 01h | 1 | RW | IER_DLH | Divisor Latch High Byte Register (DLH) |
| 01h | 0 | RW | IER_DLH | Interrupt Enable Register (IER) |
| 02h | X | RO | IIR_FCR | Interrupt Identification Register (IIR) |
| 02h | X | WO | IIR_FCR | FIFO Control Register (FCR) |
| 03h | X | RW | LCR | Line Control Register |
| 04h | X | RW | MCR | Modem Control Register |
| 05h | X | RO | LSR | Line Status Register |
| 06h | X | RO | MSR | Modem Status Register |
| 07h | X | RW | SPR | Scratch Pad Register |

§ §

# 19 Serial Peripheral Interface (SPI)

The SoC implements a Serial Peripheral Interface (SPI) controller as the interface to the SPI Flash image storage. The SPI controller supports a maximum of two SPI NOR Flash devices and supports frequencies of 17 MHz (default), 30 MHz, and 48 MHz. The SPI controller also supports one discrete TPM device.

The SoC SPI NOR Flash controller allows the Flash component(s) to be shared by various masters including BIOS, Intel® ME, LAN Controllers (GbE), and Innovation Engine (IE). The Flash controller is accessed from either the host, IE, or Intel® ME root space. The GbE accesses to Flash (for both the controller and software driver) are via the GbE controller MMIO space.

Related reference material:

- LAN Controller (GbE) related material described in this section can be reference to Chapter 13, "LAN Controllers".
- Intel® ME related material described in this section can be reference to Chapter 36, "System Management".
- For Innovation Engine (IE) related material described in this chapter, see Chapter 37, "Innovation Engine", see Table 1-5, "Reference Documents" on page 127.

**Figure 19-1. What is Covered in This Chapter**

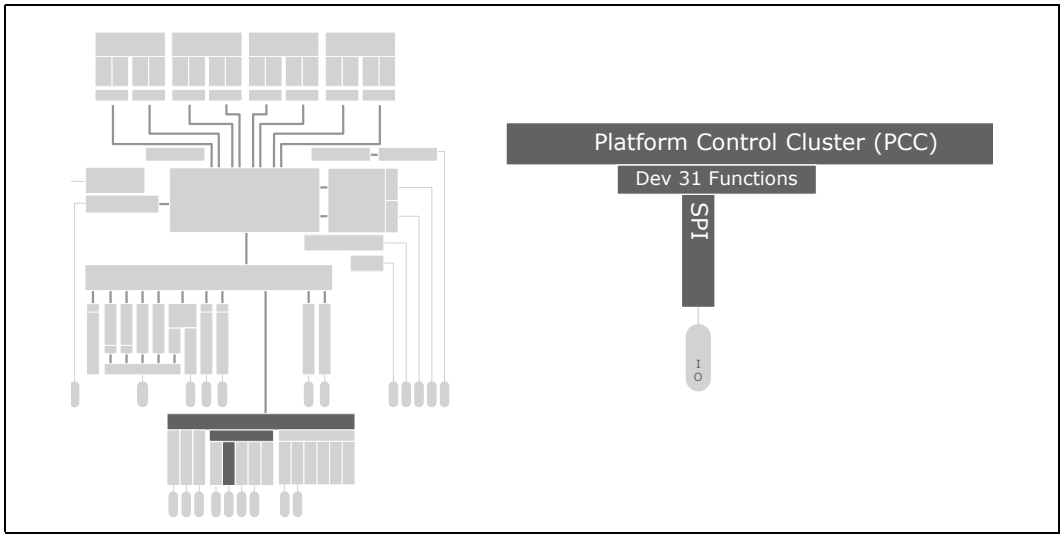



**Table 19-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| SFDP 1.0, JESD216 | -- | May 2014 | *Serial Flash Discoverable Parameters (SFDP), JEDEC Specification JESD216B* |

## 19.1 Signal Descriptions

The signal descriptions are shown in Table 19-2 For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction column of Table 19-2 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.

### Table 19-2. Signal Names and Descriptions

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| SPI_CLK | O | Yes | **SPI Clock:** The SPI clock signal during idle the bus owner will drive the clock signal low. |
| SPI_MOSI_IO[0] | I,O | Yes | **SPI Master OUT Slave IN:** Data output pin. |
| SPI_MISO_IO[1] | I,O | Yes | **SPI Master IN Slave OUT:** Data input pin. |
| SPI_IO[3:2] | I,O | Yes | **Extended SPI** I/O**:** to 4-bit data. |
| SPI_CS_N[0] | O | Yes | **SPI Chip Select 0:** Used as the SPI bus request signal for the Flash device. |
| SPI_CS_N[1] | O | Yes | **SPI Chip Select 1:** Used as the SPI bus request signal for the Flash device. |
| SPI_TPM_CS_N | O | Yes | **SPI Trusted Platform Module (TPM) Chip Select:** Used as the TPM dedicated request signal.<br>***Note:*** The "TPM Over SPI or LPC Select" Soft Strap much be set to SPI to use TPM on the SPI interface. |
| Boot BIOS Strap | I | Yes | See Chapter 4, "Strapping and Configuration." |
| LPC Select | I | Yes | See Chapter 4, "Strapping and Configuration." |
| Flash Security Override | I | Yes | See Chapter 4, "Strapping and Configuration." |
| Top Swap Override | I | Yes | See Chapter 4, "Strapping and Configuration." |

## 19.2 Feature List

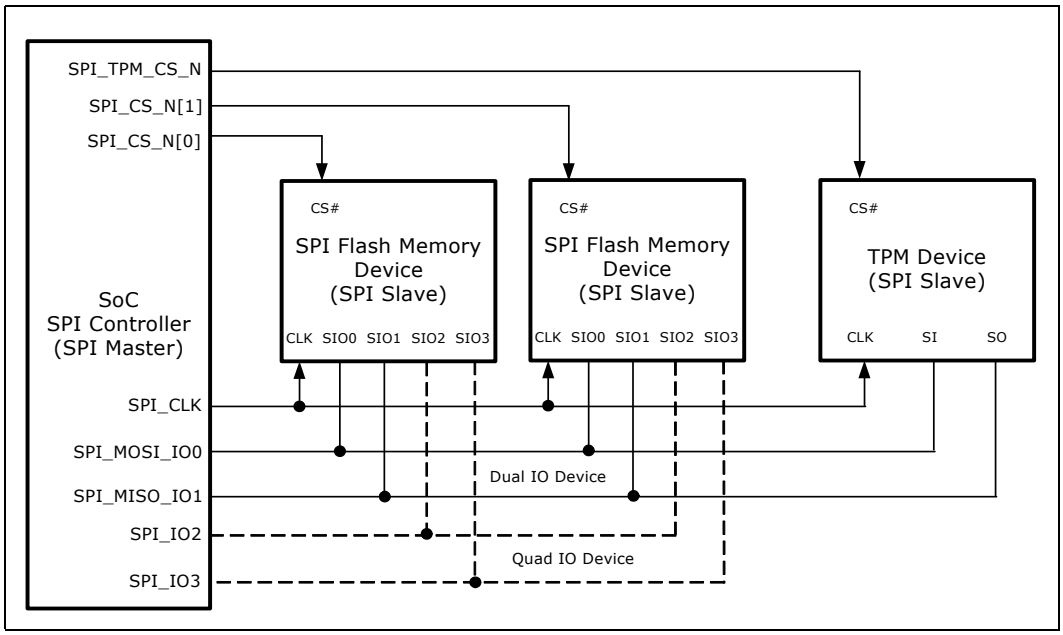
- Support for up to two SPI Flash devices
- The maximum size of each Flash device supported is determined by the SFDP-discovered addressing capability of each device.
  - Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing.
  - Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing
- Different size SPI Flash devices may be used on each chip select.
- Both SPI Flash devices must be from the same vendor and family.
- Supports five configurable protecting ranges.
- SoC soft-strap information is supported.
- 3.3V SPI I/O buffer VCC.
- Supports the SPI fast read instruction and frequencies 17 MHz (default), 30 MHz, and 48 MHz.
- Support for SPI attached TPM with one TPM chip select (SPI_TPM_CS_N).
- Supports up to quad mode.

## 19.3 Architectural Overview

Communication on the SPI bus is done with a master–slave protocol. See Figure 19-2 for the master-slave connection of the SPI devices. The Slave Output (SO) data is implemented through a tri-state bus.

The SPI Flash controller allows the Flash component(s) to be shared by various masters including the BIOS, Intel® ME, Innovation Engine (IE), and LAN Controllers. The Flash controller is accessed from either the host or Intel® ME or IE root space.

**Figure 19-2. Connection to the SPI Devices**




The chip select 0 (the SPI_CS_N[0] pin) Flash device must have a valid descriptor for SoC to boot.

### 19.3.1 Operation Mode

The SPI controller must operate in the descriptor mode with optional security access restrictions.

#### 19.3.1.1 Descriptor Mode

The descriptor mode is required to enable many features:

- Supports Flash that is divided into regions and accessible by multiple masters.
  a. Regions are allowed to extend across multiple Flash components.
  b. Regions are aligned to 4K blocks/sectors.
- The soft strap region provides the ability to use Flash NVM as an alternative to hardware pull-up/pull-down resistors for the SoC.
- The Flash descriptor contains the Flash upper map.
  a. This is used by software to define Flash vendor specific capabilities.
- The top 256B of the Flash descriptor is reserved for use by the OEM.

#### 19.3.1.2 Security

- Descriptor-based region restriction: Hardware enforced security restricting master accesses to different regions.
  a. Flash descriptor region settings define separate read/write access to each region per master.
  b. Flash security override SoC hard (pin) strap to remove all descriptor-based security (does not affect register-based protected ranges).
  c. Masters can grant other masters read/write access to their region.
- Protected range registers
  a. Five sets of lockable protected range registers that can restrict program register accesses.
  b. Can span multiple regions.
  c. Separate read and write protection.
- SMI write protection for the BIOS
  a. If enabled, will cause an SMI if a program register access occurs when enabled. The primary purpose of this requirement is to support SMI-based BIOS update utilities.
- Illegal instruction protection for instructions such as chip erase.
- Lockable software sequencing opcodes, disabled by default.

#### 19.3.1.3 SPI Flash Access

- Direct read access
- Program register access
  a. Hardware sequencing: Sequencing uses hardware (HW) to provide the basic instructions of read, write, and erase.
  b. Software sequencing: Allows software (SW) to use only a legal opcode.
- Support for the boot BIOS on the SPI.
  a. SoC hard (pin) straps (BBS) determine the BIOS boot location.
- Pre-fetching/caching to improve performance
  a. Separate 64B pre-fetch/cache for the host.

### 19.3.1.4 Flash Component Requirements

- Serial Flash Discoverable Parameter (SFDP) support.
  Refer to *Serial Flash Discoverable Parameters (SFDP), JEDEC Specification JESD216B*

***Notes:***
1. SPI controller does not support legacy non-SFDP devices. SFDP support and an SFDP table are required for flash components.
2. SPI controller collects SFDP information for each chip select (CS0 and CS1) and will use the highest supported features for each chip select. CS0 and CS1 do not have to use the same modes.

- Supports two SPI Flash components using two separate chip select pins, SPI_CS_N[0] and SPI_CS_N[1].
- The Flash controller hardware allows selection of either 64k or 4k erase opcodes if the devices advertise these opcodes via SFDP.
- The maximum size of Flash supported is determined by the SFDP-discovered addressing capability of each device. Each component can be up to 16 MB (32 MB total addressable) using 3-byte addressing. Each component can be up to 64 MB (128 MB total addressable) using 4-byte addressing.
- 3.3V SPI I/O buffer VCC
- Supports frequencies 17 MHz (default), 30 MHz, and 48 MHz (48 MHz support requires components that meet 66 MHz timing).
- Supports the SPI fast read instruction.
- Supports the SPI dual output and dual I/O fast read instruction.
- Supports the SPI quad output and quad I/O fast read.
- Uses the standardized Flash instruction set.
- Supports Flash sizes which are not 2^N (where N is an integer), also known as non-power of 2, with the following restrictions:
  a. The Flash regions must be programmed to the actual size of the Flash component(s).
  b. If using two Flash components, the first Flash component (the one with the Flash descriptor) must be of size 2^N. The second Flash component can be a non-power of 2 size. If using only one Flash component, it can be of non-power of 2 size.
  c. The value programmed in the Flash descriptor component density must be set to the next power of 2 value larger than the non-power of 2 size.

*Note:*

**If** the SoC is configured to support "Flash dual output and dual I/O fast read" and/or the "quad output and quad I/O fast read" **and** The SPI flash device advertises via *Serial Flash Discoverable Parameters* (SFDP) the "dual output and dual I/O fast read instruction" and/or the "quad output and quad I/O fast read instruction".

**Then:** In addition to SFDP, the SPI flash device must also **enable** these SFDP advertised dual mode and/or quad mode at the SPI flash device power on. Check with your SPI flash vendor to determine if a specific device supports this requirement.

**Else:** The SoC will not boot

| SoC | Dual (DORE/DIORE) | | Quad (QORE/QIORE) | |
|---|---|---|---|---|
| Flash SFDP | Yes | | Yes | |
| SPI Flash Enabled at power on | No | Yes | No | Yes |
| Boot | No | Yes | No | Yes |

## 19.3.2 Flash Regions

The Flash is divided into regions. The following regions are assigned the following content.

**Table 19-3. Flash Regions**

| Region | Name |
|---|---|
| **0** | **Descriptor** |
| **1** | **BIOS** |
| **2** | **Intel® ME** |
| 3 | Reserved |
| **4** | **Platform Data Region** |
| 5 | Reserved |
| 6 | Reserved |
| 7 | Reserved |
| 8 | Reserved |
| 9 | Reserved |
| **10** | **Innovation Engine (IE)** |
| **11** | **LAN Controller 0** |
| **12** | **LAN Controller 1** |
| 13 | Reserved |
| 14 | Reserved |
| 15 | Reserved |

The regions have the following characteristics:

- Region 0, 1 (SPI boot only), and 2 are required to bring up the SoC. Other regions may be optional to include in the Flash, and depends on how the feature is used.
- Region 0 must be located in the first sector of component 0 and SPI_CS_N[0] and all other regions can be organized in any order.
- Region boundaries must be aligned to erasable sector boundaries, with a minimum granularity of 4 KB.
- Regions can extend across multiple components.

**Figure 19-3. Flash Regions**

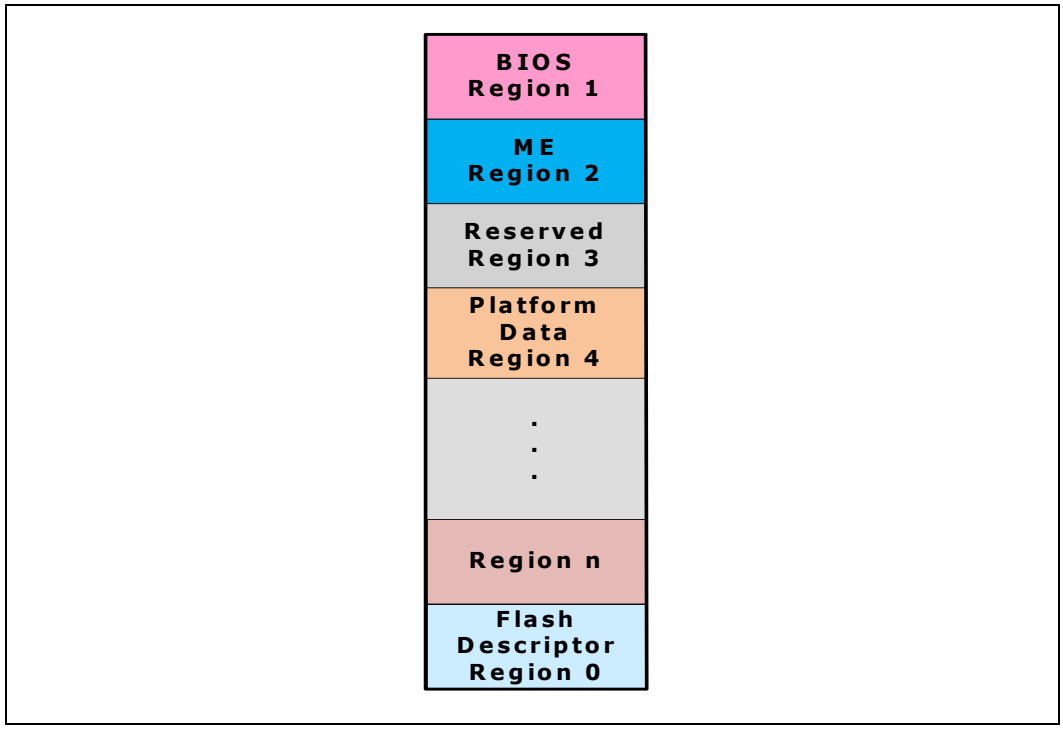
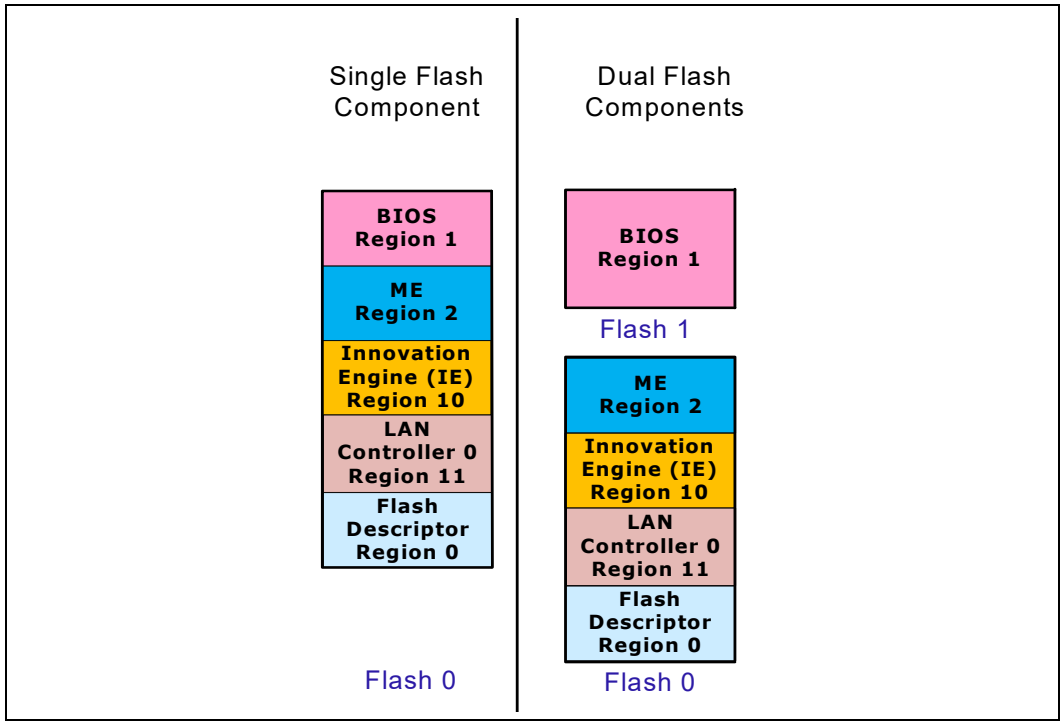



**Figure 19-4. Organize Flash Region Single Flash vs. Dual-Flash Components**



### 19.3.2.1 Flash Component Boundary

Flash regions are allowed to cross the component boundary, except for the LAN controller region.

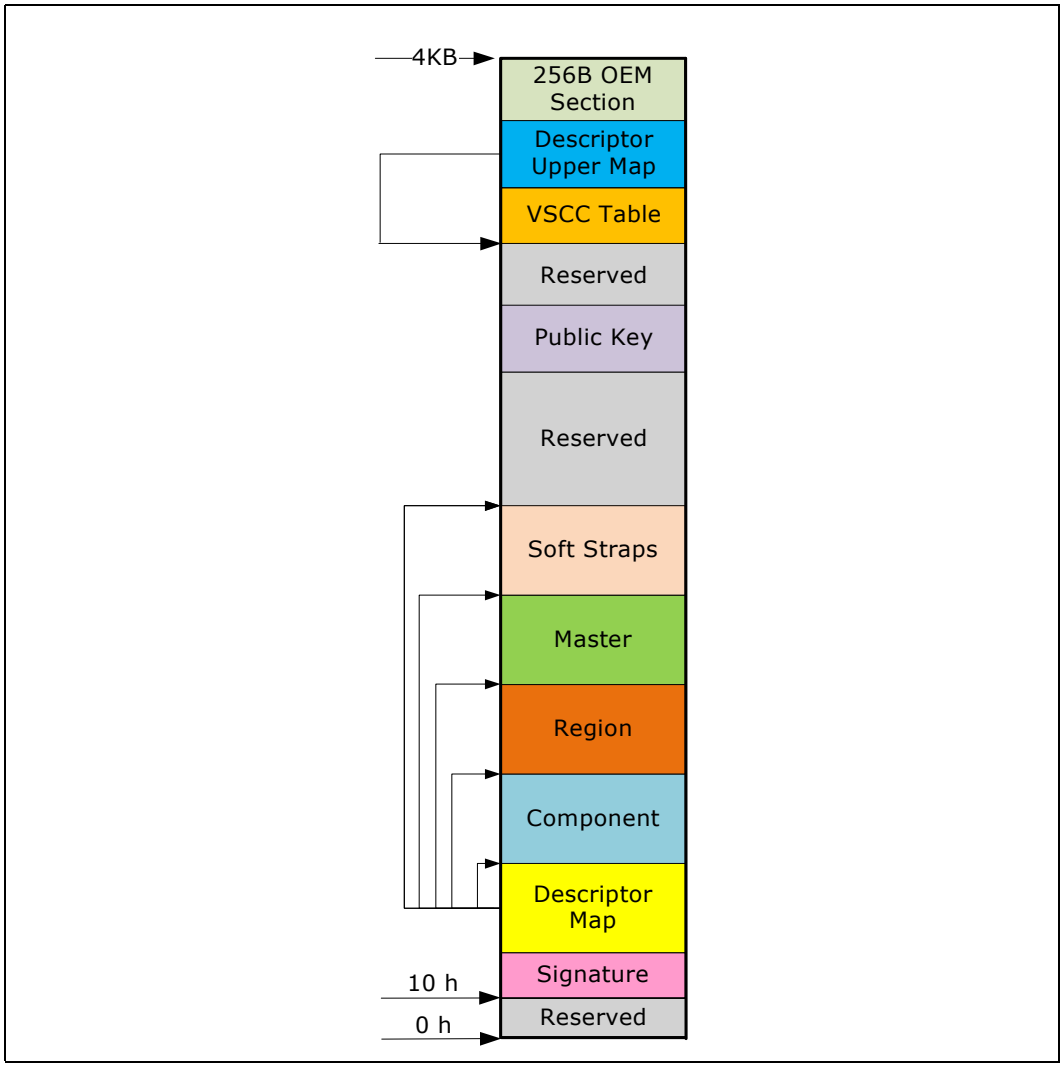
### 19.3.2.2 Flash Descriptor Region 0

The bottom sector of Flash component 0 contains the Flash Descriptor. The maximum size of the Flash Descriptor is 4K, even if the flash device's sector size is larger than 4K. (The hardware is not required to perform any checks to enforce this -- it's simply a descriptor programming error to configure it any other way.).

The information stored in the Flash descriptor is only written during the board-level manufacturing process. Intel requires the read/write permissions be set to read only when the system containing the SoC leaves the manufacturing floor.

The Flash descriptor is made up of the sections indicated in Figure 19-3.

**Figure 19-5. Flash Descriptor Sections**



- The **Reserved** sections are reserved.
- The **Signature** section selects the descriptor mode and verifies if the Flash is programmed and functioning. This signature is used to identify if the Flash is valid. The Flash signature is located at offset 10h.
- The **Descriptor Map** section defines the logical structure of the Flash in addition to the number of components used.
- The **Component** section has information about the SPI Flash in the system including:
  - Density of each component
  - Illegal instructions (such as chip erase)
  - Frequencies for read, fast read, and write/erase instructions
- The **Region** section points to the four other regions and the size of each region.
- The **Master** region contains the security settings for the Flash, grants read/write permissions for each region, and identifies each master by a requester ID.
- The **Soft Straps** section contains parameter bits that are used to configure the SoC features and/or behaviors.
- 1 KB is used for **Public Key** storage. The key programmed by the OEM.
- The **VSCC Table** section holds the JEDEC ID and the Vendor Specific Component Capabilities (VSCC) information of the entire SPI Flash supported by the NVM image.
- The **Descriptor Upper Map** section determines the length and base address of the **VSCC Table** section.
- The **OEM Section** is 256-bytes reserved at the top of the Flash descriptor for use by an OEM. The information stored by the OEM can only be written during the manufacturing process as the Flash descriptor read/write permissions must be set to read only when the computer leaves the manufacturing floor. The SoC Flash controller does not read this information. Security permissions in the Flash descriptor should be set to provide read access to both the SoC and for the Intel® ME so they read the OEM section and other Flash descriptor runtime as needed.

*Note:* It is the user's responsibility to allocate sufficient space in each descriptor region to hold the requested amount of data. Hardware does not check for overlaps in the descriptor map.

## 19.4 Flash Access

The two types of Flash accesses are:

- Direct access
- Program register access

### 19.4.1 Direct Access

- The SoC cores, Intel® ME, and IE are the only agents that can do direct read access. Access is limited to reading from their primary region.
- The SPI controller does not support direct writes to Flash, no master can do a direct write to the SPI Flash. Write accesses must take place through the Programmed Register interface.
- Direct write is supported to the SPI TPM when it is enabled.

#### 19.4.1.1 BIOS Direct Read in Descriptor Mode

The SPI Flash controller only maps host direct reads to Flash. BIOS can issue Programmed Register accesses independent of the Boot BIOS Strap allowing BIOS to use the SPI Flash as NVM storage concurrently with LPC.

At Boot, BIOS can access the SPI Flash controller by doing a direct read to the 16MB address range just below 4GB, FF00_0000h to FFFF_FFFFh. BIOS can reduce the size of the 16MB of address space by writing to the BIOS Decode Enable (BDE) register. Independent of the settings of the BIOS Decode Enable Offset, the SPI Controller will not claim any direct reads that exceed the size of the active BIOS region. That is, the BIOS Decode Enable can reduce the memory space that is accessible via a direct read, but it cannot be used to address Flash exceeding the size of the active BIOS region.

When the SPI Flash controller sees a read request from the SoC Core to the BIOS Decode region it will claim the transaction. The read request will be a DW aligned address. The SPI Flash controller calculates the Flash Linear Address (FLA) from the BIOS address by subtracting the difference between the BIOS upper range (FFFF_FFFC) and the actual BIOS address, from the Flash Region Limit:

FLA[26:0] <= (Flash_Regionn_Limit[26:12] & "FFCh") - (FFFF_FFFCh - bios_address[31:2] & "00b") where n=1 if primary BIOS is enabled.

If the BIOS access is to the Legacy F segment (F0000h - FFFFFh) or Legacy E segment (E0000h - EFFFFh) which ranges must be aliased to just below 4GB, the SPI controller will calculate the FLA using the following arithmetic[1]:

FLA[26:0] <= (Flash_Regionn_Limit[26:12] & "FFCh") - (FFFF_FFFCh - "FFF" & bios_address[19:2] & "00b") where n=1 if primary BIOS is enabled.

The Flash Linear Address must be converted to a Flash Physical Address by the SPI Flash Controller prior to issuing on the SPI interface.

If the Top Swap (TS) bit is set when booting from SPI Flash, the SPI Controller will invert one of A16, A17, A18, A19 or A20 (note that LPC TS is only A16) depending on the value of the SPI - Boot Block Size soft strap. The inversion only occurs for the top most memory block, i.e. when all bits above the inverted bit are '1'. The inverted address is then used in the equation above to determine the Flash Linear Address. The Top Swap functionality does not apply to accesses generated to the holes below 1MB. The SPI Controller performs the address bit inversion on only the Direct Memory Read access method.

1. Intel is not validating legacy support.

#### 19.4.1.2 Chip Select and Flash Linear Address (FLA)

The controller manages two chip select pins SPI_CS_N[0] and SPI_CS_N[1].

The 27-bit Flash Linear Addresses (FLA[26:0]) either calculated from the direct read address or as programmed in the Flash Linear Address program register must be converted into a 26 bit Flash Physical Address (FPA[25:0]) prior to being issued on the SPI pins according to the following algorithm.

The size of the Flash component is always taken from density field in the SFDP table. FLCOMP.C0DEN is no longer used by the SPI Controller.

If FLA <= Sizeof(Flash Component 0) then

FPA[25:0] = FLA[25:0];

Chip Select = 0

Else

FPA[25:0] = FLA[26:0] - Sizeof(Component 0)

Chip Select = 1

When a 3-byte address is issued the controller sends FPA[23:0] and truncates FPA[25:24].

### 19.4.2 Programmed Register Access

- Any master attempts to read or write the SPI programmed registers and SoC internal initiator logic check does not match that of the Primary Region master, an Unsupported Request will be returned.
- Program register access uses hardware or software sequencing. Section 19.10, “SPI Support for Trusted Platform Module (TPM)” on page 617.
- For programmed register accesses, the SPI controller implemented hardware security to ensure that the master never issues a read, write, or erase request to any region that it does not have proper access to by comparing the programmed FLA with the base and limit of each of the regions and verifying that master as read or write access to the specified address. Attempted access to addresses which do not fall within any defined region shall cause the access permission check to fail.
- Any attempt to access a region without proper access permissions will prevent the cycle from running and will cause the Flash Cycle Error (HSFSTS_CTL.FCERR) register error bit to be set.

## 19.5 Flash Security and Protection

The controller provides additional Flash protection mechanisms beyond the access restrictions defined in the Flash descriptor:

- Range write protection
- Intel® ME write protected range
- IE write protected range
- SMI-based global write protection (BIOS writes/erases to the BIOS regions)

The mechanisms are conceptually ORed together such that if any of the mechanisms indicate that the access should be blocked, then it is blocked.

The SoC provides these protections in hardware. It is critical that the hardware must not allow malicious software to modify the address or opcode pointers after determining that a cycle is allowed to run such that the actual cycle that runs on SPI should have been blocked.

If the command associated with an atomic cycle sequence is blocked according to the SoC configuration, the SPI controller must not run any of the sequence.

A blocked command will appear to software to finish, except that the FCERR status bit is set in this case.

**Table 19-4. Flash Protection Mechanism Summary**

| Mechanism | Accesses Blocked[1] | Range Specific | Reset- Override or SMI-Override |
|---|---|---|---|
| BIOS Range Write Protection | Reads/Writes by Host | Yes | Reset Override |
| Intel® ME Write Protected Range | Writes by any Master | Yes | Reset Override |
| Intel® ME Protected Range | Reads/Writes by ME | Yes | Reset Override |
| IE Protected Range | Reads/Writes by IE | Yes | Reset Override |
| SMI Write Protect | Writes by Host | Applies to both BIOS regions | SMI Override |

1. "Writes" implies both writes and erases

### 19.5.1 Flash Range Write and Read Protection

The SPI controller provides a method for blocking writes and reads to specific ranges in the SPI Flash when the protected ranges are enabled. This is achieved by checking the read or write cycle type and the address of the requested command against the base and limit fields of a read or write protected range. The protected range applies to all master-initiated Flash accesses except the BIOS. Therefore, those register protections do not apply to hardware-initiated descriptor reads. The BIOS protected range registers only apply to BIOS accesses, the IE protected range registers only apply to IE accesses, etc.

The range specified in the Flash Range registers are allowed to span any addresses, independent of whether that master has read or write access to the region(s) in, or partially in, the protected address range.

*Note:* Once software has locked down the BIOS or Host Protected Range registers, this mechanism remains in place until the next partition reset.

### 19.5.2 SMI Based Global Write Protection

The SPI controller blocks BIOS writes to the BIOS regions of the SPI Flash when the Write Protect Disable bit is cleared (i.e., protected). For software sequencing, this is achieved by checking the opcode type information (which can be locked down by the initial boot BIOS) of the requested command. Hardware sequencing write protection is triggered when attempting to issue a write or erase command.

The BIOS Write Protect bit does not apply to any other masters that have the write access permission to the BIOS regions. Host CPU accesses to the regions are not protected by the BIOS Write Protect bit. A blocked write may also generate an SMI. See Section 19.10, "SPI Support for Trusted Platform Module (TPM)" on page 617.

### 19.5.3 Flash Security Override Strap

The Flash descriptor security override strap is an SoC hard (pin) strap. The purpose of the SoC hard (pin) strap is to override all descriptor-based protections, including region definitions and master region permissions. It does not override register-based or software- programmable protections. It is typically used when an OEM wants to re-write a Flash image during manufacturing or board refurbishment. The effects of the override are limited to the following:

- The master region read access and master region write access permissions that were loaded from the Flash descriptor master section will be overridden giving every master read and write permissions to the entire Flash component(s) including areas outside the defined regions.
- The address checks to ensure that a linear address falls within the total size of the Flash components are not performed.
- No RequesterID security checks are performed

*Note:* The software programmable Protected Range registers, e.g., host PR0-PR4, GPR0, and IE PR0-PR4, are not affected by the Flash descriptor security override strap.

- The region boundaries for direct reads are still enforced.
- The Flash controller determines the size of the Flash devices from SFDP, not the descriptor. Therefore, cycles may be sent to SPI_CS_N[0] or SPI_CS_N[1] independent of the security override strap.

*Note:* Any access that exceeds the size of the device attached to SPI_CS_N[0] is sent to SPI_CS_N[1], independent of whether a Flash device is connected to SPI_CS_N[1].

- The SoC hard (pin) strap does not change the behavior of the BC.WPD register.

The Flash descriptor security override SoC hard (pin) strap resides in the resume power well. When the strap is enabled, security override will be performed by the Flash controller even though core power well may be down.

### 19.5.4 Illegal Instructions

Any attempt to execute an illegal instruction will prevent the cycle from running and will cause the FCERR register error bit to be set.

## 19.5.5 Error Status Table

Table 19-5 summarizes the status bits that are set under different error scenarios. The "cycle in progress" and "scenario" columns refer to the same master, i.e. it's only an error if Master A performs the scenario while Master A has a cycle in progress.

**Table 19-5. Programmed Access Error Status**

| Cycle in Progress | Scenario | HW Seq FCERR | HW Seq DONE | SW Seq FCERR | SW Seq DONE |
|---|---|---|---|---|---|
| Hardware Sequence | Writes to HSFSTS_CTL byte_address 6h | 1 | N/A | N/A | N/A |
| | Writes to HSFLCTL.07h | 1 | N/A | N/A | N/A |
| | Writes to SSFLCTL.A1h (with SCGO =1) | N/A | N/A | 1 | |
| | Writes to SSFLCTL.A1h (with SCGO =0) | N/A | N/A | 1 | 0 |
| | Writes to SSFLCTL.A2h | N/A | N/A | 1 | 0 |
| | Writes to SSFLCTL.A3h | N/A | N/A | 1 | 0 |
| | Writes to FADDR or FDATAx | 1 | N/A | N/A | N/A |
| | | | | | |
| Software Sequence | Writes to HSFLCTL.06h (with FGO =1) | 1 | 1 | N/A | N/A |
| | Writes to HSFLCTL.06h (with FGO = 0) | 1 | 0 | N/A | N/A |
| | Writes to HSFLCTL.07h | 1 | 0 | N/A | N/A |
| | Writes to SSFLCTL.A1h | N/A | N/A | 1 | N/A |
| | Writes to SSFLCTL.A2h | N/A | N/A | 1 | N/A |
| | Writes to SSFLCTL.A3h | N/A | N/A | 1 | N/A |
| | Writes to SSFLCTL.A3h | N/A | N/A | 1 | N/A |
| | Writes to FADDR or FDATAx | N/A | N/A | 1 | N/A |

## 19.5.6 Region Checks

Reads and writes are not allowed to cross 4k address boundaries in Flash. This implies they are not allowed to cross devices.

Region definitions must be contiguous, i.e., no unallocated Flash addresses between any two regions. Unallocated Flash addresses between the highest region limit and the end of Flash are allowed. It is a descriptor programming error to define a region that extends beyond the total size of Flash.

## 19.6 SMI Generation

### 19.6.1 Synchronous SMI

The SPI controller sends synchronous SMI assert/de-assert message pairs when the Write Protect Disable bit is written from 0 to 1 while the Lock Enable bit is set. Both bits are in the BIOS Control register in host root space PCI header. The SPI controller waits for the smi_ack message before sending the completion for the non-posted write which set the bit. The SPI controller sends the de-assert synchronous SMI message when software clears the SPI_SYNC_SS bit.

### 19.6.2 Asynchronous SMI

The SPI controller sends asynchronous SMI assert/de-assert message pairs when configured to generate an SMI for the hardware and software sequencing Flash cycle done. The SMI for Flash cycle done is enabled via HSFSTS_CTL.FSMIE for hardware sequencing and SSFSTS_CTL.SME for software sequencing. The BIOS Control(BC).SE_BWP bit does not control SMI for Flash cycle done. If the SMI is enabled, assert SMI is sent when the Flash cycle done transitions from 0 to 1. If the SMI is enabled, de-assert SMI is sent when the Flash cycle done transitions from 1 to 0 or when the SMI enable becomes false.

BIOS writes/erases may be blocked. When ASE_BWP = 1, such a blocked write or erase attempt by the host to the BIOS region(s) will cause an asynchronous SMI even if the transaction would also be blocked due to multiple causes. De-assert SMI is sent when the Flash cycle done transitions from 1 to 0 or when ASE_BWP becomes false.

## 19.7 Hardware vs. Software Sequencing

Hardware and software sequencing are the two methods used to communicate with the Flash via the programming registers.

Hardware sequencing is the safer method for communication with the flash. Since the SPI controller uses SFDP to determine the capabilities of each attached SPI flash, hardware sequencing will only use those supported capabilities. Software sequencing on the other hand might use capabilities that the SPI flash doesn’t support causing unpredictable result. Software sequencing is by default disabled in the SoC. To enable it please use SS9 bit [13] SPI - Host Software Sequencing Enable.

### 19.7.1 Hardware Sequencing

Hardware sequencing has a predefined list of opcodes with only the erase opcode being programmable. This mode is only available if the descriptor is present and valid.

Table 19-6 contains a list of commands and the associated opcodes that an SPI-based serial Flash device must support to be compatible with hardware sequencing.

**Table 19-6. Hardware Sequencing Opcodes**

| Instruction Op-Code | Description |
|---|---|
| 03h | Read data |
| 05h | Read status |
| 0Bh | Fast read |
| 06h | Write enable |
| 04h | Reserved |
| 02h | Write data/program data |
| 50h | Enable the Write Status register. |
| 01h | Write Status register |
| 3Bh/Discoverable | Dual output fast read<br>Discoverable opcodes are obtained from each component SFDP table. |
| Discoverable | Dual I/O fast read |
| Discoverable | Quad output fast read |
| Discoverable | Quad I/O fast read |
| Discoverable | 4k erase. Uses the value from SFDP (if available) or else the value from the VSSC.EO register.<br>Industry standard is 20h. |
| Discoverable | 64k erase. Industry standard is D8h. |
| Discoverable | RPMC Op1. Not supported via hardware sequencing in SPT. Industry standard is 9Bh. |
| Discoverable | RPMC Op2. Uses fast read timing with 8 wait states. Not supported via hardware sequencing in SPT. Industry standard is 96h. |
| 5Ah | Read SFDP. Uses fast read timing with 8 wait states. |
| 9Fh | Read JEDEC ID. |
| B7h | Enable 32-bit addressing mode. |

If there is a Hardware Sequencing Cycle In Progress and an attempt is made to program any of the control, address, or data register the cycle is blocked and the FCERR bit is set.

If hardware sequencing allows selection of the Write Status opcode, then the Flash controller will insert a Write Enable (06h) opcode prior to issuing the Write Status. The two operations will be issued as an atomic sequence.

## 19.7.2 Software Sequencing

Any required command not supported by Hardware Sequencing must be programmed by the software in the Software Sequencing Control, Flash Address, Flash Data, and Opcode Configuration registers. The software must issue either a read ID or read JEDEC ID, or a combination of the two to determine what Flash component is attached. Based on the read ID, the software determines the appropriate opcode instruction sets to set in the program registers and at what SPI frequency to run the command.

The software must program the Flash linear address for all commands, even for those commands that do not require an address such as the read ID or read status. This is because the SPI controller uses the address to determine which chip select to use.

The opcode type and data byte count fields determine how many clocks to run before deasserting the chip enable. The Flash data is always shifted in for the number of bytes specified, and the Flash data out is always shifted out for the number of data bytes specified.

A status bit indicates when the cycle has completed on the SPI port allowing the host to know when read results are checked and/or when to initiate a new command.

The controller also provides the atomic cycle sequence for performing erases and writes to the SPI Flash. When this bit is 1 (and the Go bit is written to 1), a sequence of cycles is performed on the SPI interface without allowing other SPI devices to arbitrate and interleave cycles to the Flash device. In this case, the specified cycle is preceded by the Prefix command (8-bit programmable opcode) and followed by repeated reads to the Status register (opcode 05h) until bit 0 indicates the cycle has completed. The hardware does not attempt to check that the programmed cycle is a write or erase.

If a programmed access is initiated (Cycle Go written to 1) while the SPI controller is already busy with a direct memory read, then the SPI host hardware holds the new programmed access pending until the preceding SPI access completes.

Once the SPI controller has committed to running a programmed access, subsequent writes to the programmed cycle registers that occur before it has completed do not modify the original transaction and result in the assertion of the FCERR bit. The software never purposely behaves in this way and relies on this behavior. However, the FCERR bit provides basic error reporting in this situation. Writes to the following registers cause the FCERR bit assertion in this situation:

- Software Sequencing Control register
- Software Sequencing Address register
- SPI Data register

With the exception of illegal opcodes, the SPI controller does not police which opcodes are valid to be used in software sequencing. For example, if the software programs a dual-output-fast-read opcode, then the dual-output-fast-read cycle is issued, independent of whether the Dual-output Fast Read Enable bit was set in the component descriptor section. However, disabled opcodes for dual and quad reads will be run on the SPI pins as single-input, single-output commands, and will therefore return incorrect data to the requester and cause bus contention when the Flash device drives multiple pins of the SPI bus. Similarly, SPI controller does not use the device capabilities discovered via SFDP to alter any command programmed by software.

**SW sequencing registers:** HSFSTS_CTL, SSFSTS_CTL, PREOP_OPTYPE, OPMENU, FADDR, and FDATA. See Table 19-9, "BIOS Flash Program Register Address Map" on page 624.

- HSFSTS has the FLOCKDN bit which locks (prevents writing) most SW Sequencing opcode registers. Trusted firmware (BIOS or ME code) configures these registers, then sets FLOCKDN prior to handing control to operating system software.
- SSFSTS indicates when a transaction has completed and provides error status for the transaction. Software must write-to-clear the FCERR and Cycle Done Status bits prior to starting a transaction. Software polls this register, waiting for Cycle Done Status, before accessing read data or setting up a new transaction.
- SSFCTL contains all configuration data for a transaction, e.g., number of data bytes, which opcode(s), whether an atomic sequence of opcodes, and SPI_CLK frequency. Software sets the register SPI Cycle Go bit to initiate a transaction.
- PREOP is programmed with two Write Enable opcodes used by Flash devices. The selected PREOP opcode is sent to Flash prior to the selected OPMENU opcode when the atomic cycle sequence flag is set in the SSFCTL register. Flash devices require the Write Enable opcode be sent prior to Write and Erase instructions. Some devices also require Write Enable prior to a Write Status Register opcode. OPTYPE identifies whether each opcode in OPMENU is a read or a write cycle and whether an address must be sent following the opcode. For example, a Read Status opcode is a read instruction with no address, a Read SFDP opcode is a read instruction with an address, and a Write Status Register opcode is a write instruction with no address. Software must ensure that OPTYPE is programmed correctly for each opcode in OPMENU - hardware cannot detect or correct mismatches.
- OPMENU is programmed with the opcodes that are available for SW sequencing operations. Opcodes typically set in this register include Read SFDP (5Ah), Read JEDEC ID (9Fh), and Write Status Register (01h). Programming "read flash array" type instructions (e.g. Fast Read, Read Dual I/O, Read Quad I/O) is typically not required because the Flash controller will automatically select the most efficient read opcode when either HW Sequencing or direct reads are used.
- FADDR is programmed with the Flash Linear Address for the SW sequenced operation. FADDR must be programmed for every SW sequenced transaction because the Flash controller uses the address to determine which chip select to activate when running the transaction on the SPI bus.
- FDATA is programmed with data to be written to Flash. The Flash controller stores data read from Flash in FDATA.

Example: Read 16 bytes of SFDP from component 1 using SW sequencing

- Verify that HSFSTS.FLOCKDN=0, i.e. the SW sequencing registers are unlocked
- Write to SSFSTS to clear FCERR and Cycle Done Status
- Program OPTYPE 0 to "address required; read cycle type"
- Program OPMENU 0 to 5Ah
- PREOP is not required prior to Read SFDP; no need to touch this register
- Program FADDR to component 1, address offset zero. The size of the Flash component is always taken from density field in the SFDP table. FLCOMP.C0DEN is no longer used by the SPI Controller.
- Program SSFCTL to select opcode and start transaction. SSFCTL fields: SCF=100b, SME=0, DS=1, DBC=0xF, COP=0, SPOP=don't care, ACS=0, SCGO=1
- Poll SSFSTS, waiting for SCIP=0 to show completed SW Sequenced transaction
- Check error status in SSFSTS.FCERR
- Read 16 bytes from FDATA[3:0]

## 19.8 Serial Flash Device Compatibility Requirements

A variety of serial Flash devices exist in the market. For a serial Flash device to be compatible with the SPI bus, it must meet the minimum requirements.

### 19.8.1 BIOS SPI Flash Requirements

The SPI Flash device must meet the following minimum requirements when used explicitly for the system BIOS storage. Refer to Section 19.3.1.4, "Flash Component Requirements" on page 597.

- The erase size has at least one of the following: 64 Kbytes, or 4 Kbytes.
- The device must support multiple writes to a page without requiring a preceding erase cycle (refer to Section 19.5.1, "Flash Range Write and Read Protection" on page 605).
- The Serial Flash device must ignore the upper address bits such that an address of FF_FFFFh aliases to the top of the Flash memory for 3-byte addressing support.
- SPI compatible mode 0 support (the clock phase is 0 and data is latched on the rising edge of the clock).
- If the device receives a command that is not supported or incomplete (less than 8 bits), the device must complete the cycle gracefully without any impact on the Flash content.
- An Erase command (page, sector, block, chip, etc.) must set all bits inside the designated area (page, sector, block, chip, etc.) to 1 (Fh).
- Status register bit 0 must be set to 1 when a write, erase, or write to a status register is in progress and cleared to 0 when a write or erase is NOT in progress.
- Devices requiring the Write Enable command must automatically clear the write enable latch at the end of data program instructions.
- Byte write must be supported. The flexibility to perform a write between 1 byte to 64 bytes is recommended.

## 19.9 SPI Flash Interface

Communication on the SPI bus is done with a Master - Slave protocol. Typical bus topologies call for a single SPI Master with a single SPI Slave. The basic SPI interface consists of a four wire interface: clock (CLK), master data out (Master Out Slave In (MOSI)), master data in (Master In Slave Out (MISO)) and active low chip selects (CS0#).

The SoC SPI Flash controller supports 3 chip selects, SPI_CS_N[0], SPI_CS_N[1], and SPI_TPM_CS_N. The SPI_TPM_CS_N is dedicated to support TPM on SPI. The Flash controller supports up to 4 I/O pins, i.e. SPI_MOSI_IO[0], SPI_MISO_IO[1], SPI_IO[2] and SPI_IO[3] to comprehend the support for the Dual I/O and Quad I/O operation. The Flash controller does not support use of two or four wires for the opcode phase of a Flash transaction, e.g. 2-2-2 or 4-4-4 mode reads.

### 19.9.1 Single-Input, Dual-Output Fast Read

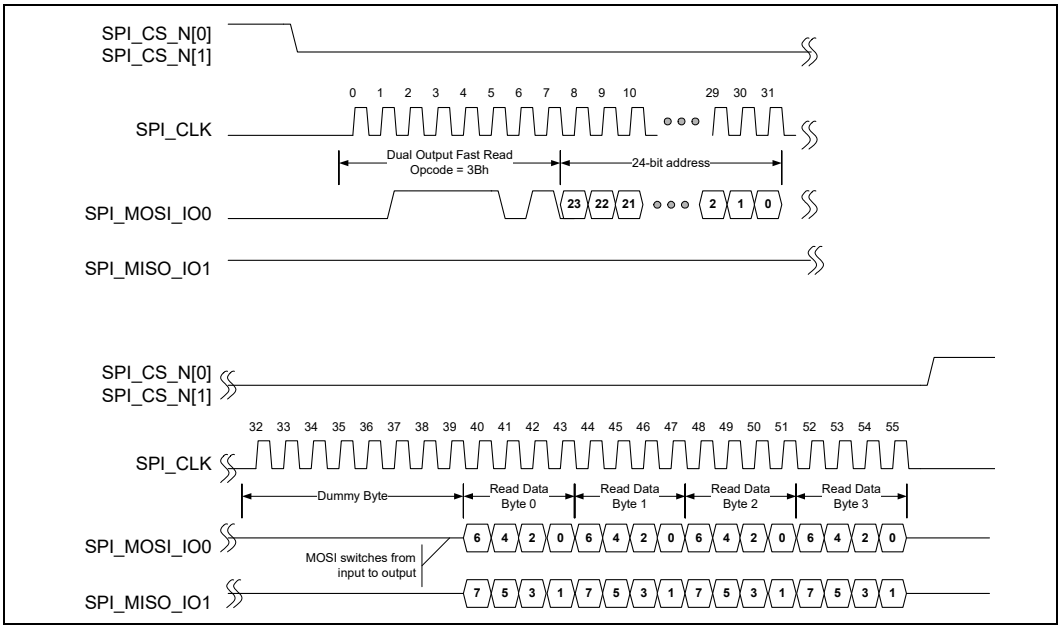
The SPI controller supports the functionality of a single-input, dual-output fast read: opcode 3Bh. This instruction has the same timing (including a dummy byte) and the same frequencies as the fast read instruction, with the difference that the read data from the Flash is presented on both the MISO and MOSI pins. During a dual-read instruction, the odd data bits are on the MISO pin and the even data bits are on the MOSI pin.

*Note:* When the dual-output-fast-read support is enabled the fast-read support must be enabled as well.

*Note:* The Micronix* SPI Flash uses a different opcode for dual-fast read and requires during the address phase the address bits are sent on both MOSI and MISO. The SoC does not support this implementation of the protocol.

*Note:* When in the 32-bit addressing mode, the address bits in the waveform in Figure 19-6 will be extended to 32 bits (i.e., 4 bytes) instead of 24 bits.

**Figure 19-6. Dual-Output-Fast-Read Timing**

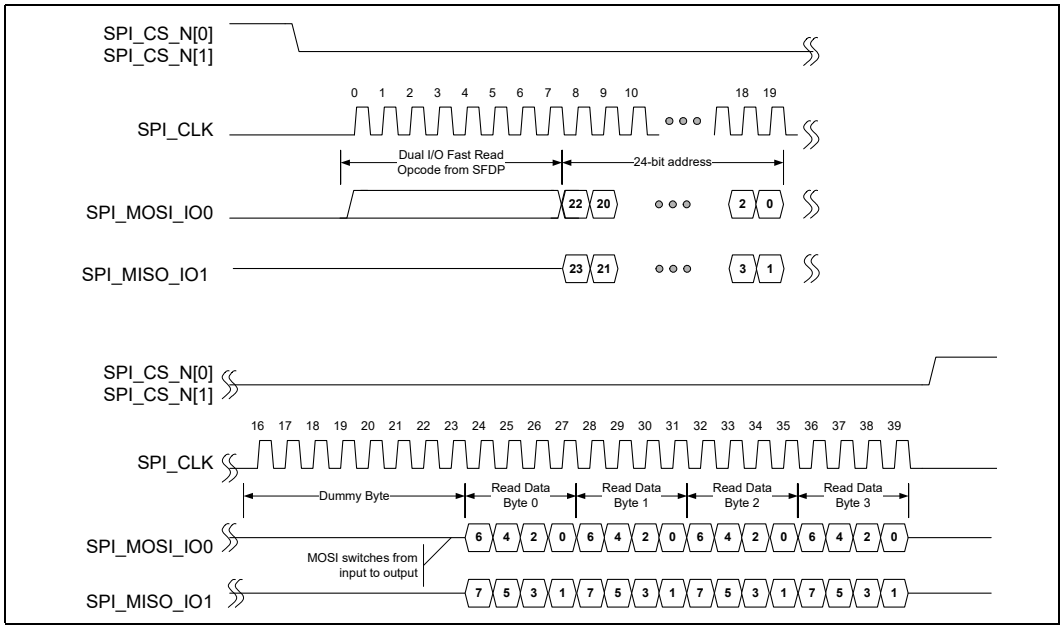


## 19.9.2 Dual-I/O Fast Read

Besides having the read data be presented on both the MOSI and MISO pins, dual-I/O fast read allows the address bits to be sent on both MOSI and MISO as well. Odd address bits are on the MISO pin and the even address bits are on the MOSI pin. The dummy byte is also on both the MOSI and MISO pins.

*Note:* When in the 32-bit addressing mode, the address bits in the waveform in Figure 19-7 will be extended to 32 bits (i.e. 4 bytes) instead of 24 bits.

**Figure 19-7. Dual-I/O-Fast-Read Timing**

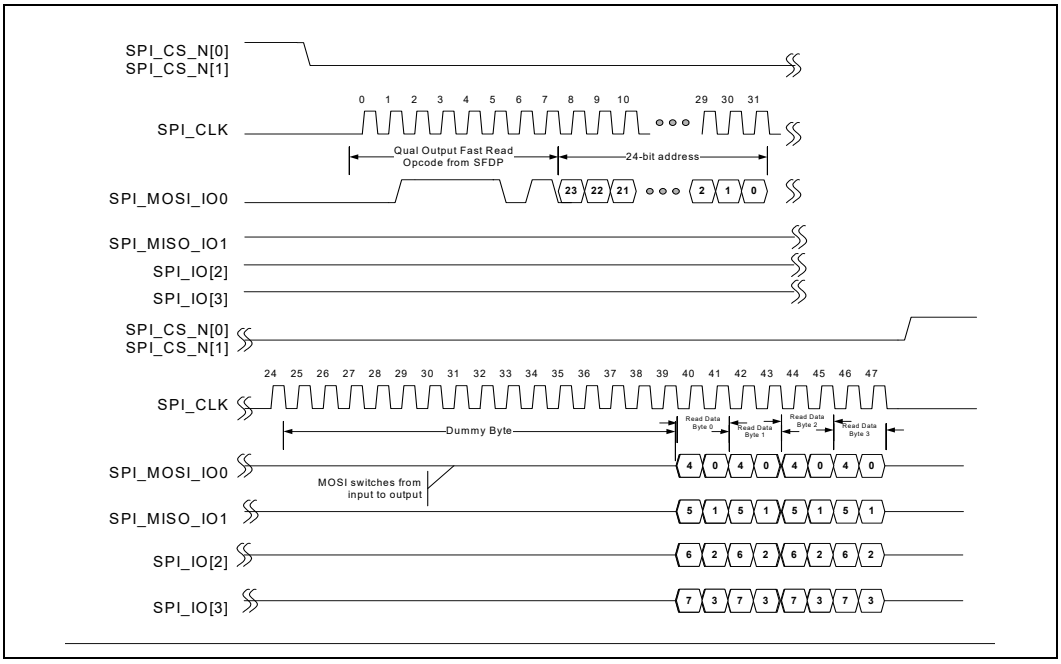


## 19.9.3 Quad-Output Fast Read

Quad-output fast read allows the read data from the Flash be presented on the MOSI, MISO, SPI_IO2, and SPI_IO3 pins, at the rate of 4 data bits per clock.

*Note:* When in the 32-bit addressing mode, the address bits in the waveform in Figure 19-8 will be extended to 32 bits (i.e. 4 bytes) instead of 24 bits.

**Figure 19-8. Quad Output Fast Read Timing**

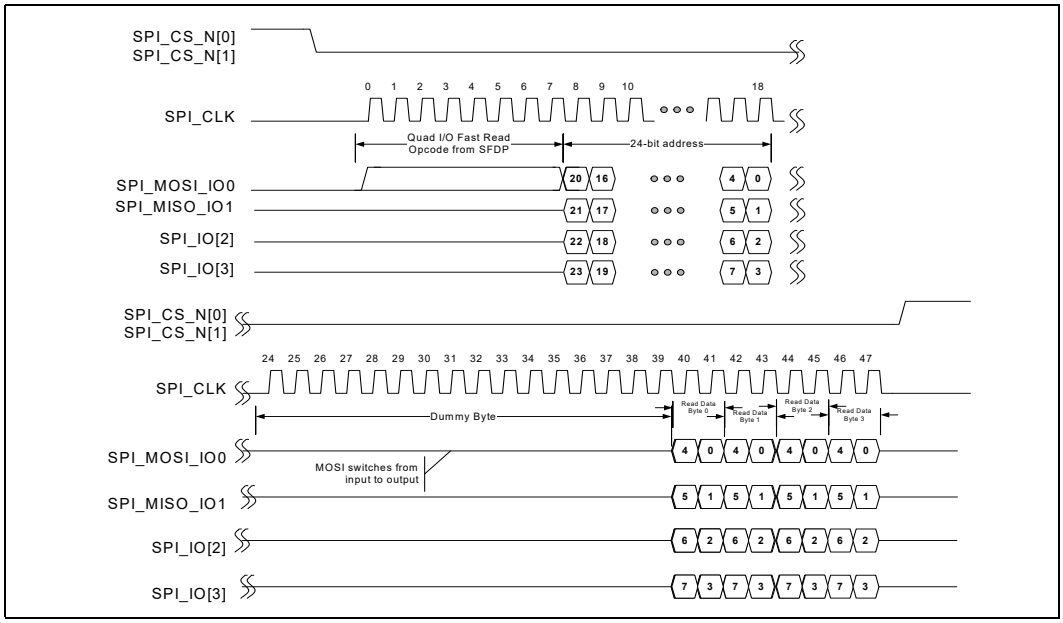


## 19.9.4 Quad-I/O Fast Read

Quad-I/O fast read allows the address, the dummy byte and the read data be presented on the MOSI, MISO, SPI_IO2 and SPI_IO3 pins, thus achieving 4 bits per clock during both the address and data phases.

When in the 32-bit addressing mode, the address bits in the waveform in Figure 19-9 will be extended to 32 bits (i.e. 4 bytes) instead of 24 bits.

**Figure 19-9. Quad-I/O-Fast-Read Timing**




## 19.9.5 JEDEC ID

Since each serial Flash device may have unique capabilities and commands, the JEDEC ID is the necessary mechanism for identifying the device so the uniqueness of the device is comprehended by the controller (master). The JEDEC ID uses the opcode 9Fh and a specified implementation and usage model. This JEDEC standard manufacturer and device ID read method is defined in Standard JESD21-C, PRN03-NV.

*Note:* Once software has locked down the BIOS or Host Protected Range registers, this mechanism remains in place until the next partition reset.

## 19.10 SPI Support for Trusted Platform Module (TPM)

The SoC supports one discrete TPM device with SPI_TPM_CS_N. The SPI controller decodes the TPM read/write directly. There is no register programming model for the TPM.

The requirements for the support are:

- The platform must have no more than 1 TPM. Therefore, the SoC must route all TPM transactions either to LPC or SPI depending on the configuration bits. Some platforms may be without TPM.
- A chip select for the SPI TPM (SPI_TPM_CS_N)
  - When the soft strap sets the TPM on SPI, the TPM chip select must only assert on the decode of TPM transactions.
  - TPM chip select cannot assert based on any programmed registers or other SW mechanism.
- The SPI controller supports accesses to the SPI TPM at whatever frequency the SPI bus has been configured.
- The TPM requires the support for the interrupt routing. However, the TPM interrupt pin is routed to the SoC. Thus, the TPM interrupt is completely independent from the SPI controller.
- The SPI controller is configurable to prevent TPM access when the descriptor is invalid (or no Flash is attached). This is a security requirement for the SoC. When configured to prevent TPM access when the descriptor is invalid, the Flash controller behaves as if the TPM on the SPI does not exist on the system.

### 19.10.1 TPM Protocol on SPI

Though the SPI controller supports 26-bit addressing on SPI, only 24-address bits are sent to the TPM. The FEh byte is dropped. For example, an incoming address of FED4_4024h would be sent on the SPI as D4_4024h.

Access to the SPI TPM is always with single-address input and single output at the single data rate. Dual-output, dual-I/O, quad-output and quad-I/O operations are not supported on SPI_TPM_CS_N with the TPM.

The following rules apply for the data transfer:

- Data is shifted Most Significant Bit (MSB) first, Least Significant Byte (LSB) first.
- The address and command are shifted MSB first for the entire field.
  - The 24-bit address shifts A23 first, then A22, A21..., A0.
- Master and slave both drive data on the falling edge of the clock.
- The TPM accesses always get a 24-bit address that is the offset from FE00_0000h.
  - Asserting the SPI_TPM_CS_N is an indication that the SoC did a full decode and the cycle is in the FED4_xxxxh range.
- Only SPI mode 0 is supported (CPHA = 0, CPOL = 0).

It is legal to transmit any number of bytes from 1 to 64. Zero length reads or writes are not allowed. If the transfer is less than 4B, then the corresponding bits would be left out. For instance, on a 2B write the last transfer on MOSI would be Data[8] (LSB of MSB).

There is no status byte for the transfer. If the write to the TPM failed, the TPM would not honor what was received, and SW/driver will understand the SW command did not succeed and perform the appropriate recovery mechanisms. If the read fails, then the SoC would return all FFs for the data, which would signal a failure to the driver.

## Table 19-7. Address and Data Transfer for TPM

| Bit Transfer Order on MISO/MOSI | BYTE on MISO/MOSI | Usage | Notes |
|---|---|---|---|
| | 67 for 64B transaction<br>11 for 8B transaction | Reserved | |
| 57-63 – last bits on wire | 7 | Data[30:24] | |
| 56 | | Data[31] | MSB of fourth LSB |
| 49-55 | 6 | Data[22:16] | |
| 48 | | Data[23] | MSB of third LSB |
| 41-47 | 5 | Data[14:8] | |
| 40 | | Data[15] | MSB of second LSB |
| 33-39 | 4 | Data[6:0] | |
| 32 | | Data[7] | MSB of LSB |
| **Optional flow control can be done in this window. This is the only place in the bit transfer where flow control can be done.** | | | |
| 31 | 3 | Addr[0] | LSB of address |
| 9-30 | 1-3 | Addr[22] down to Addr[1] | |
| 8 | 1 | Addr[23] | MSB of address |
| 2-7 | 0 | Bits [5:0]<br>Size of transfer where bit [5] of this field is the third bit transferred on the wire, and bit [0] is the eighth bit on the wire. This field is a 0s-based count of the bytes. Any byte count from 1 to 64 is legal. | Bit [5:0] decode<br>11_1111 = 64 bytes<br>(For 63 down to 6 bytes)<br>00_0100 = 5 bytes<br>00_0011 = 4 bytes<br>00_0010 = 3 bytes<br>00_0001 = 2 bytes<br>00_0000 = 1 byte |
| 1 | | bit [6]. Reserved | |
| 0-first bit on wire | | Byte0, bit [7]. Read/Write | 1 = read<br>0 = write |

### 19.10.2 TPM Flow Control on SPI

Note: This description uses idealized, zero-delay timing for illustrative purposes.

For the TPM operation, it involves HW generating transactions directly to the TPM with minimum or no SW involvement. As such, there needs to be a simple flow control mechanism on SPI since HW can't poll busy bits or use other SW mechanisms. Therefore the following flow control is allowed by the TPM.

SPI protocol doesn't have a defined flow control mechanism. Thus a new flow control mechanism is being created for the TPM on SPI.

The flow control is on a transaction basis and not on a byte basis. For example, a read or write to the data register can be at most 4B in length today, moving to 8B or 64B in the future. The TPM will accept the write data when it has the full size of buffer available to be written (1-64B), or provide the read data when it has the full amount of data (1-64B) ready to deliver, again based on the size of the transaction. The overhead of allowing flow control between each byte is too high with almost no benefit.

Since the specification allows for larger sizes of transactions in the future, the SoC will have limited, if any, HW checking on accesses to the TPM address space. If the SoC receives transaction for any size from 1B to 64B that doesn't cross a 64B boundary, it must issue that transaction on SPI as received. The SoC must accept all transactions of any length on any address boundary to FED4_0000h to FED4_4FFFh, as long as they don't cross a 64B boundary.

The TPM transaction on SPI consists of 1B of command, 3B of address, followed by write data from the SoC or read data from the TPM. The TPM may insert wait states after the 4B of command and address have been received.

The mechanism to insert wait states is as follows. For a read to the TPM, the command and address are driven on MOSI and the TPM responds with data on MISO. With no wait states, the TPM would drive data on the next falling clock edge after the falling clock edge that the SoC drove the last address bit. The flow control mechanism added for the TPM is that the SoC will monitor the MISO pin in the same clock window that A[0] (the last address bit) is valid.

The TPM receives the address, where address bit[2] is captured a clock and a half before it has to drive the flow control bit. For reads, the registers are on a 4B (or 8B) boundary, so that bit[2] of the address should be sufficient to determine if the register is available or not to be read. Then bit[1] and [0] are only used to steer the correct bytes to the bus, but shouldn't be needed to determine flow control.

*Note:* Having byte-based flow control allows the transaction layer to handle the retransmit of the data. This allows for faster clock speeds. It also keeps all transfers on a byte basis and not having a few bits in between. It provides plenty of time for the TPM to decode the address and determine if it is ready to receive the data or not. This allows faster SPI clocking to speed up the transfers.

There is no further flow control allowed. Once the data starts coming from the TPM, it must provide the entire transactions worth of data, which can be from 1 to 64 bytes. In this example, if the SoC had latched a '1' on the 5th rising clock, then it would start latching in data starting on the 6th rising edge, which is the normal behavior without wait states. The TPM may insert any number of wait states that it needs.

If the TPM intends to drive wait states, it should drive MISO low when it detects CS# asserted. Once the TPM determines the address and whether it can accept or return the data, it drives MISO high for one clock in the wait state window.

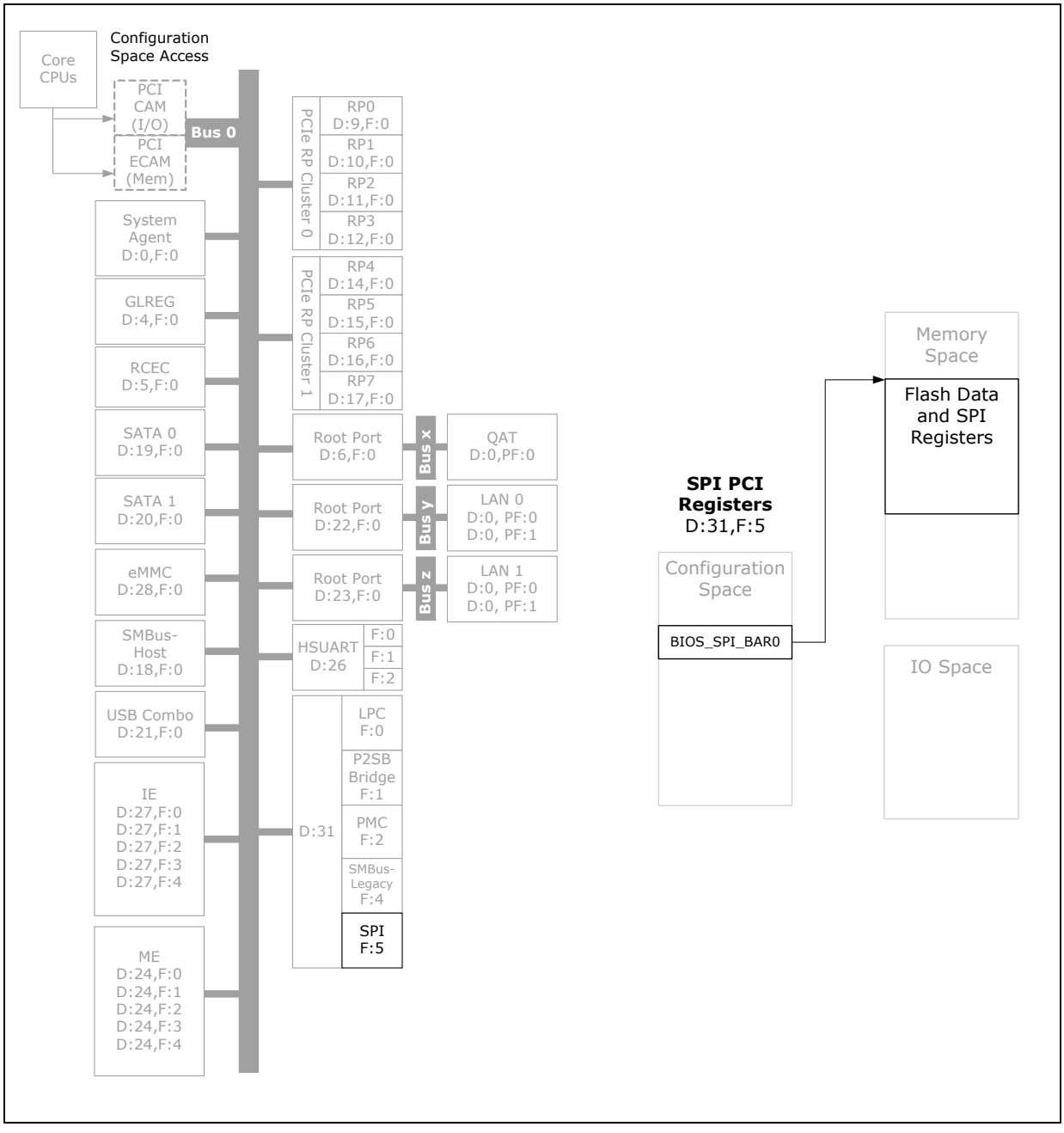
*Note:* The wait state is defined as '0' on MISO, so that if there is no TPM present at all, then the design has a weak pull-up on MISO. Having a pull-up means that the SPI controller sees a '1', which indicates no wait states, and will latch in FFh on the read. This follows standard master abort behavior of FFh for read data and matches the behavior when the TPM was on LPC.

For writes, the mechanism is similar. The SoC will latch the value of MISO on the 5th rising edge. This is the same edge that the last address bit is latched by the slave. In other words, the slave drives MISO to '0' in the same clock that the master drives the last bit of the address. The master samples MISO every 8 clocks after that and will only continue when it samples a '1' during this bit position. If MISO is '0' to request a wait state, then the data during that byte is not valid and the TPM must drop the data. If there is a wait state, the master will drive invalid data over and over until the slave stops requesting wait states. The SoC will sample MISO on the last data bit of the byte (multiples of 8 clocks after the first wait state window). Again, the TPM must hold MISO as '0' for 8 clocks each time it requests a wait state. If MISO is '1', this indicates the TPM is ready for the entire write, and the SoC will drive the 1st and subsequent data bytes on the following clocks. If MISO is still '0', the SoC will continue sending invalid data until it receives a '1' on MISO in the wait state window. Note that the flow control only applies to the first byte and not between each byte. Once the data starts transmitting, the entire write data will be sent with no further flow control. An example for a 2B write that has 2 wait states would be, drive command and address for 4 bytes, then invalid data byte, invalid data byte, valid data byte 0, valid data byte 1.

## 19.11 Register Map

The SPI controller is a standalone PCI device in the host root space at bus 0, device 31 (decimal), and function 5. The registers listed in Figure 19-10 are described in detail.

**Figure 19-10.Register Map**



### 19.11.1 PCI Configuration and Capabilities

The SPI controller also has fixed mappings, independent of the PCI configuration header, in the ranges:

- 4G - 16M to 4G -- direct reads to the BIOS regions. This depends on the boot interface. These could go to LPC.
- Legacy E -- Reads and Writes to E segment are to DRAM only not the BIOS regions
- Legacy F -- Reads and Writes to F segment are to DRAM only not the BIOS regions
- 0xFED4_xxxx -- TPM direct reads and writes if the TPM over the SPI is enabled.
- The registers marked no in the SPI supported column behave as Read Only (RO) and return zero.

**Table 19-8. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D31:F5 (decimal) | Register Short Name | Description | SPI Supported |
|---|---|---|---|
| 0x0 | BIOS_SPI_DID_VID | Device ID and Vendor ID | Yes |
| 0x4 | BIOS_SPI_STS_CMD | Status and Command | Yes |
| 0x8 | BIOS_SPI_CC_RID | Revision ID | Yes |
| 0xC | BIOS_SPI_BIST_HTYPE_LT_CLS | BIST, Header Type, Latency Timer, Cache Line Size | No |
| 0x10 | BIOS_SPI_BAR0 | SPI BAR0 MMIO | Yes |
| 0x28 | BIOS_SPI_CCP | Cardbus CIS Pointer | No |
| 0x2C | BIOS_SPI_SID_SVID | Subsystem and Vendor ID | No |
| 0x30 | BIOS_SPI_XRBAR | Expansion ROM Base Address | No |
| 0x34 | BIOS_SPI_CAPP | Capabilities List Pointer | No |
| 0xD0 | BIOS_SPI_UR_STS_CTL | SPI Unsupported Request Status | Yes |
| 0xD8 | BIOS_SPI_BDE | BIOS Decode Enable | Yes |
| 0xDC | BIOS_SPI_BC | BIOS Control | Yes |

## 19.11.2 Memory Mapped Registers

These registers are mapped in the host root space. The address offsets are relative to BIOS_SPI_BAR0. Reads to unimplemented addresses return 0s.

**Table 19-9. BIOS Flash Program Register Address Map (Sheet 1 of 2)**

| Base Address | MEM Address | Instance Name | Name |
|---|---|---|---|
| BIOS_SPI_BAR0 | 0x0 | BIOS_BFPREG | SPI BIOS MMIO PRI |
| BIOS_SPI_BAR0 | 0x4 | BIOS_HSFSTS_CTL | Hardware Sequencing Flash Status and Control |
| BIOS_SPI_BAR0 | 0x8 | BIOS_FADDR | Flash Address |
| BIOS_SPI_BAR0 | 0xC | BIOS_DLOCK | Discrete Lock Bits |
| BIOS_SPI_BAR0 | 0x10 | BIOS_FDATA0 | Flash Data 0 |
| BIOS_SPI_BAR0 | 0x14 | BIOS_FDATA1 | Flash Data 1 |
| BIOS_SPI_BAR0 | 0x18 | BIOS_FDATA2 | Flash Data 2 |
| BIOS_SPI_BAR0 | 0x1C | BIOS_FDATA3 | Flash Data 3 |
| BIOS_SPI_BAR0 | 0x20 | BIOS_FDATA4 | Flash Data 4 |
| BIOS_SPI_BAR0 | 0x24 | BIOS_FDATA5 | Flash Data 5 |
| BIOS_SPI_BAR0 | 0x28 | BIOS_FDATA6 | Flash Data 6 |
| BIOS_SPI_BAR0 | 0x2C | BIOS_FDATA7 | Flash Data 7 |
| BIOS_SPI_BAR0 | 0x30 | BIOS_FDATA8 | Flash Data 8 |
| BIOS_SPI_BAR0 | 0x34 | BIOS_FDATA9 | Flash Data 9 |
| BIOS_SPI_BAR0 | 0x38 | BIOS_FDATA10 | Flash Data 10 |
| BIOS_SPI_BAR0 | 0x3C | BIOS_FDATA11 | Flash Data 11 |
| BIOS_SPI_BAR0 | 0x40 | BIOS_FDATA12 | Flash Data 12 |
| BIOS_SPI_BAR0 | 0x44 | BIOS_FDATA13 | Flash Data 13 |
| BIOS_SPI_BAR0 | 0x48 | BIOS_FDATA14 | Flash Data 14 |
| BIOS_SPI_BAR0 | 0x4C | BIOS_FDATA15 | Flash Data 15 |
| BIOS_SPI_BAR0 | 0x50 | BIOS_FRACC | Flash Region Access Permissions |
| BIOS_SPI_BAR0 | 0x54 | BIOS_FREG0 | Flash Region 0 |
| BIOS_SPI_BAR0 | 0x58 | BIOS_FREG1 | Flash Region 1 |
| BIOS_SPI_BAR0 | 0x5C | BIOS_FREG2 | Flash Region 2 |
| BIOS_SPI_BAR0 | 0x60 | BIOS_FREG3 | Flash Region 3 |
| BIOS_SPI_BAR0 | 0x64 | BIOS_FREG4 | Flash Region 4 |
| BIOS_SPI_BAR0 | 0x68 | BIOS_FREG5 | Flash Region 5 |
| BIOS_SPI_BAR0 | 0x6C | BIOS_FREG6 | Flash Region 6 |
| BIOS_SPI_BAR0 | 0x70 | BIOS_FREG7 | Flash Region 7 |
| BIOS_SPI_BAR0 | 0x74 | BIOS_FREG8 | Flash Region 8 |
| BIOS_SPI_BAR0 | 0x78 | BIOS_FREG9 | Flash Region 9 |
| BIOS_SPI_BAR0 | 0x7C | BIOS_FREG10 | Flash Region 10 |
| BIOS_SPI_BAR0 | 0x80 | BIOS_FREG11 | Flash Region 11 |
| BIOS_SPI_BAR0 | 0x84 | BIOS_FPR0 | Flash Protected Range 0 |
| BIOS_SPI_BAR0 | 0x88 | BIOS_FPR1 | Flash Protected Range 1 |
| BIOS_SPI_BAR0 | 0x8C | BIOS_FPR2 | Flash Protected Range 2 |

## Table 19-9. BIOS Flash Program Register Address Map (Sheet 2 of 2)

| Base Address | MEM Address | Instance Name | Name |
|---|---|---|---|
| BIOS_SPI_BAR0 | 0x90 | BIOS_FPR3 | Flash Protected Range 3 |
| BIOS_SPI_BAR0 | 0x94 | BIOS_FPR4 | Flash Protected Range 4 |
| BIOS_SPI_BAR0 | 0x98 | BIOS_GPR0 | Global Protected Range 0 |
| BIOS_SPI_BAR0 | 0xA0 | BIOS_SSFSTS_CTL | Software Sequencing Flash Status and Control |
| BIOS_SPI_BAR0 | 0xA4 | BIOS_PREOP_OPTYPE | Prefix Opcode and Opcode Type Configuration |
| BIOS_SPI_BAR0 | 0xA8 | BIOS_OPMENU0 | Opcode Menu0 Configuration |
| BIOS_SPI_BAR0 | 0xAC | BIOS_OPMENU1 | Opcode Menu1 Configuration |
| BIOS_SPI_BAR0 | 0xB4 | BIOS_FDOC | Flash Descriptor Observability Control |
| BIOS_SPI_BAR0 | 0xB8 | BIOS_FDOD | Flash Descriptor Observability Data |
| BIOS_SPI_BAR0 | 0xC0 | BIOS_AFC | Additional Flash Control |
| BIOS_SPI_BAR0 | 0xC4 | BIOS_SFDP0_VSCC0 | Vendor Specific Component Capabilities for Component 0 |
| BIOS_SPI_BAR0 | 0xC8 | BIOS_SFDP1_VSCC1 | Vendor Specific Component Capabilities for Component 1 |
| BIOS_SPI_BAR0 | 0xCC | BIOS_PTINX | Parameter Table Index |
| BIOS_SPI_BAR0 | 0xD0 | BIOS_PTDATA | Parameter Table Data |
| BIOS_SPI_BAR0 | 0xD4 | BIOS_SBRS | SPI Bus Requester Status |
| BIOS_SPI_BAR0 | 0xE0 | BIOS_FREG12 | Flash Region 12 |
| BIOS_SPI_BAR0 | 0xE4 | BIOS_FREG13 | Flash Region 13 |
| BIOS_SPI_BAR0 | 0xE8 | BIOS_FREG14 | Flash Region 14 |
| BIOS_SPI_BAR0 | 0xEC | BIOS_FREG15 | Flash Region 15 |
| BIOS_SPI_BAR0 | 0x108 | BIOS_RPMC0_D0 | RPMC SFDP Table. Mapping the registers into MMIO space allows the option of BIOS discovering RPMC in systems. No RPMC registers are used by SPI controller. |
| BIOS_SPI_BAR0 | 0x10C | BIOS_RPMC1_D0 | RPMC SFDP Table. Mapping the registers into MMIO space allows the option of BIOS discovering RPMC in systems. No RPMC registers are used by SPI controller. |
| BIOS_SPI_BAR0 | 0x110 | BIOS_RPMC0_D1 | RPMC SFDP Table. Mapping the registers into MMIO space allows the option of BIOS discovering RPMC in systems. No RPMC registers are used by SPI controller. |
| BIOS_SPI_BAR0 | 0x114 | BIOS_RPMC1_D1 | RPMC SFDP Table. Mapping the registers into MMIO space allows the option of BIOS discovering RPMC in systems. No RPMC registers are used by SPI controller. |
| BIOS_SPI_BAR0 | 0x118 | BIOS_BM_RAP | BIOS Master Read Access Permissions |
| BIOS_SPI_BAR0 | 0x11C | BIOS_BM_WAP | BIOS Master Write Access Permissions |
| BIOS_SPI_BAR0 | 0x1B0 | BIOS_IE_PR0 | IE Flash Protected Range 0 |
| BIOS_SPI_BAR0 | 0x1B4 | BIOS_IE_PR1 | IE Flash Protected Range 1 |
| BIOS_SPI_BAR0 | 0x1B8 | BIOS_IE_PR2 | IE Flash Protected Range 2 |
| BIOS_SPI_BAR0 | 0x1BC | BIOS_IE_PR3 | IE Flash Protected Range 3 |
| BIOS_SPI_BAR0 | 0x1C0 | BIOS_IE_PR4 | IE Flash Protected Range 4 |

§ §

# 20 Low Pin Count (LPC) Controller

The LPC serves as a PCI-to-Industry Standard Architecture (ISA) bridge to devices connected to the LPC interface pins. The bridge is discovered by the software at bus 0, device 31 (decimal), function 0 in the configuration address space. This bridge also accommodates a number of integrated legacy peripherals, most of which were originally designed for the ISA.

*Note:* 3.3V must be supplied to VCC_LPC_ESPI_3P3_1P8.

**Figure 20-1. What is Covered in This Chapter**

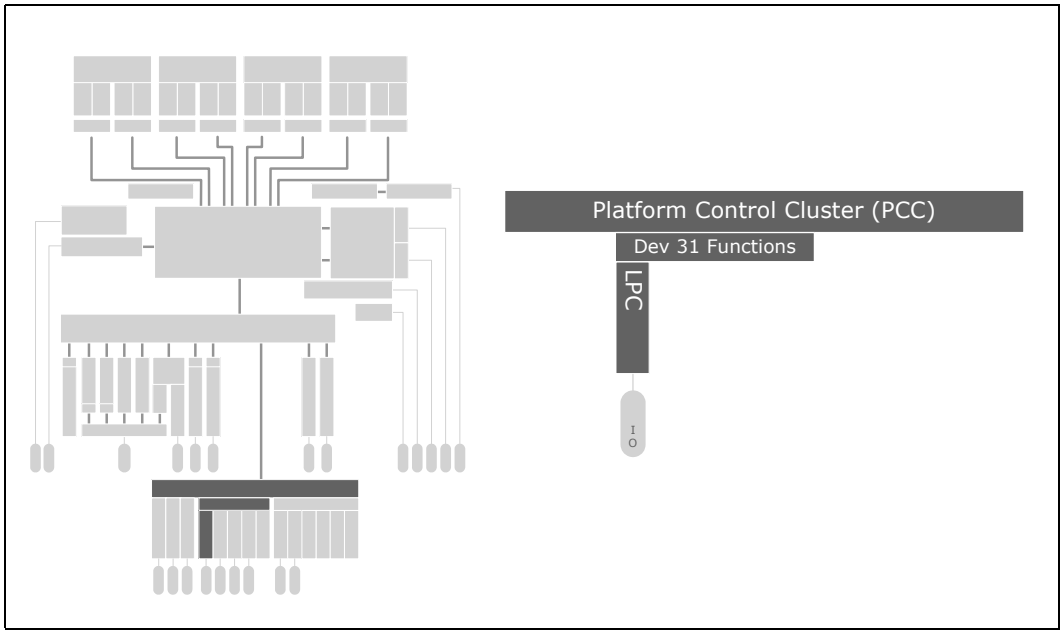



**Table 20-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| LPC Specification | 1.1 | August 2002 | *Intel® Low Pin Count (LPC) Interface Specification* |
| PCI Specification | 3.0 | February 2004 | *PCI Local Bus Specification, Revision 3.0* |

## 20.1 Signal Descriptions

The signal descriptions are shown in Table 20-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction/Type column of Table 20-2 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-Tri = Tri-state Output. SoC output circuit uses a three-state driver.

### Table 20-2. Signal Names and Descriptions

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| LPC_AD[3:0] | I,O-Tri | Yes | Multiplexed command, address, and data |
| LPC_FRAME_N | O | Yes | **Frame:** Indicates start of a new cycle and termination of a broken cycle. The SoC provides an internal 20-kΩ Pull-Up (PU) resistor on this signal pin. |
| LPC_CLKOUT[1:0] | O | Yes | **Clock:** Same 24-MHz clock for LPC bus interface. The SoC provides an internal 20-kΩ Pull-Down (PD) resistor on each of these signal pins. |
| LPC_CLKRUN_N | I,O-Tri | Yes | **Clock Run:** As an input, this active-low signal is generated by LPC peripherals that need the LPC_CLKOUT for a period of time beyond the four clocks guaranteed at the end of a transfer cycle. The SoC, as the central resource that generates the LPC_CLKOUT, monitors the LPC_CLKRUN_N signal to regulate its frequency for low-power operation.<br>As an output, the SoC provides a sustained tri-state signal. When the SoC drives the LPC_CLKRUN_N signal low (asserted), it indicates the LPC_CLKOUT is operating at its normal frequency. When driven high (de-asserted), the SoC indicates a stopped or very-slow clock frequency.<br>The platform board must provide a Pull-Up (PU) resistor to sustain a logic-high voltage when the SoC and attached LPC peripherals float this tri-state bus. |
| LPC_SERIRQ | I,O-Tri | Yes | **Serialized IRQ:** SERIRQ is an active-low sustained tri-state bus, synchronized with the LPC_CLKOUT signal. As the Serialized Interrupt Host Controller, the SoC drives this pin low for a periods of time, high for a periods of time, and other times floats this pin. The external agents attached to this pin provide the interrupt IRQ and I/O Check indications through low and high signaling, and float this line when they do not own the bus cycle. Electrically, LPC_SERIRQ conforms with the sustained tri-state parameters defined in the PCI Local Bus Specification, Revision 3.0. The platform board must provide a Pull-Up (PU) resistor to sustain a logic-high voltage when the SoC and attached agents float this tri-state bus. |
| ESPI_IRCOMP | I,O | No | **ESPI_IRCOMP:** Special compensation resistor to support 3.3V. The platform board must provide a 100 Ω, 1% resistor connected from this pin to VSS. |
| Boot BIOS Strap | I | Yes | See Chapter 4, "Strapping and Configuration." |
| LPC Decode Select | I | Yes | See Chapter 4, "Strapping and Configuration." |
| VCC_LPC_ESPI_3P3_1P8 | I (Voltage) | n/a | The platform must supply V3P30 to this voltage-supply pin. The SoC voltage-supply pins are listed in Section 39.3, "Directory of Voltage Supply and Sense Pin Names" on page 1029. Also see Chapter 4, "Strapping and Configuration". |

## 20.2 Feature List

LPC controller implements a *Intel® Low Pin Count (LPC) Interface Specification*.

## 20.3 Architectural Overview

The LPC controller implements a *Intel® Low Pin Count (LPC) Interface Specification*. The LPC bus is operating at a clock frequency of 24 MHz. The LPC interface is used for connection of various legacy components.

### 20.3.1 LPC Cycle Types

The LPC controller implements all of the cycle types described in the *Intel® Low Pin Count (LPC) Interface Specification*.

There is no support for DMA or bus mastering from the LPC.

### 20.3.2 Memory Cycle

The LPC provides a generic memory range for decoding memory cycles and forwarding as standard LPC memory cycles on the LPC bus. The generic memory decode range is 64 KB in size, locatable anywhere within the 4 GB memory space. The range is configured by the BIOS during POST to provide the memory resources needed. The BIOS will advertise the range to the operating system (OS) as reserved memory to avoid any resource conflict.

For cycles targeting firmware, firmware memory cycles are used. Only 8-bit transfers are performed. If a larger transfer appears on the backbone, the LPC controller will break it into multiple 8-bit transfers until the request is satisfied.

If the cycle is not claimed by any peripheral (and subsequently aborted), the LPC controller will return a value of all 1s to the SoC core. This is to maintain compatibility with ISA memory cycles where pull-up resistors would keep the bus high if no device responds.

### 20.3.3 I/O Cycle

All I/O cycles will be performed as 8-bit transfers per the LPC specification. If a larger size is attempted, the LPC controller will perform multiple 8-bit transfers until the request is satisfied.

### 20.3.4 DMA Cycle

DMA is not supported.

### 20.3.5 Bus Master Cycle Notes

Bus mastering to the LPC is not supported.

### 20.3.6 LFRAME# Usage

The controller follows the usage of LFRAME# as defined in the *Intel® Low Pin Count (LPC) Interface Specification*.

### 20.3.7 LPC PD# Protocol

This signal is provided to the LPC peripherals using the SoC output signal SUS_STAT_N. After driving SUS_STAT_N low (active), the SoC drives LPC_FRAME_N (LFRAME#) low, and tri-states or drives low the LPC_AD[3:0] bus.

*Note:* *Intel® Low Pin Count (LPC) Interface Specification* defines LPCPD# protocol where at least 30µs is from LPCPD# assertion to LRESET# assertion. This specification explicitly states the protocol only applies to entry or exit of low-power states which does not include asynchronous reset events. The SoC asserts both SUS_STAT_N (LPCPD#) and PMU_PLTRST_N (LRESET#) at the same time during a global reset.

### 20.3.8 LPC Interface Decoders

In order to allow I/O and memory mapped cycles to go to the LPC interface, the LPC controller includes several decoders. The decoders are located at offset 80h - 9Fh in the device configuration space.

### 20.3.9 SERR# Generation

When the LPC receives an error SYNC on the LPC bus, an internal SERR# will be generated. This occurs only if SERR reporting is enabled for the LPC.

### 20.3.10 Dynamic LPC Clock Control

This LPC clock control is achieved through the LPC_CLKRUN_N signal. If an LPC device needs the 24 MHz LPC clock, then it can assert.

#### 20.3.10.1 Verifying if Safe to Stop the LPC Clock

With a lack of LPC activity, the SoC is able to shut off the LPC clocks to conserve power. LPC activity is defined as any activity that would require the LPC clock to be running.

Any of the following conditions will indicate that it is *not* safe to stop the LPC clock:

- Cycles on the LPC (as a target)
- Cycles of any internal device that would need to go on the LPC bus (as a target)
- SERIRQ activity (as a target and master)

Behavioral description:

1. When there is a lack of activity (as defined above) for t90 clocks, the SoC will drive the LPC_CLKRUN_N signal high t91 later.
2. One clock later, the SoC will tri-state the LPC_CLKRUN_N signal.

#### 20.3.10.2 Conditions for Maintaining the LPC Clock

LPC devices that wish to maintain the LPC clock running will observe the LPC_CLKRUN_N signal high, and then must drive it low within t92 clocks.

Behavioral description:

1. When the SoC has driven the LPC_CLKRUN_N signal high, it will then check to see if it is driven low (externally).
2. After observing the LPC_CLKRUN_N signal low for 1 clock, the SoC will then start driving the signal low.
3. If an internal device needs the LPC bus, then the SoC will drive LPC_CLKRUN_N the clock signal active.

### 20.3.10.3 Conditions for Stopping the LPC Clock

If no device drives LPC_CLKRUN_N low once it has been high for t93 clocks, the SoC will stop the LPC clocks.

### 20.3.10.4 Conditions for Restarting the LPC Clock

Behavioral description:

1. If a peripheral asserts the LPC_CLKRUN_N signal, the SoC will restart the clock.
2. The SoC will drive LPC_CLKRUN_N low 1 clock after the first rise edge of the LPC clock. The peripheral likewise must continue to drive LPC_CLKRUN_N low until the second rising edge of the LPC clock is detected.
3. If an internal source requests the clock to be restarted, the SoC will assert LPC_CLKRUN_N.

**Figure 20-2. LPC_CLKRUN_N Timings**

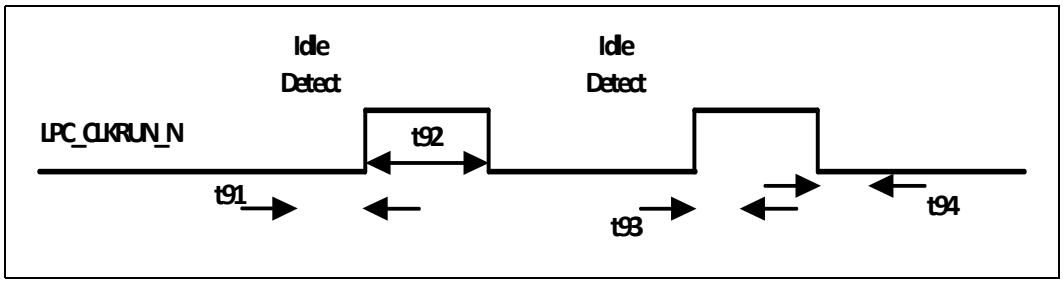



**Table 20-3. LPC_CLKRUN_N Timings**

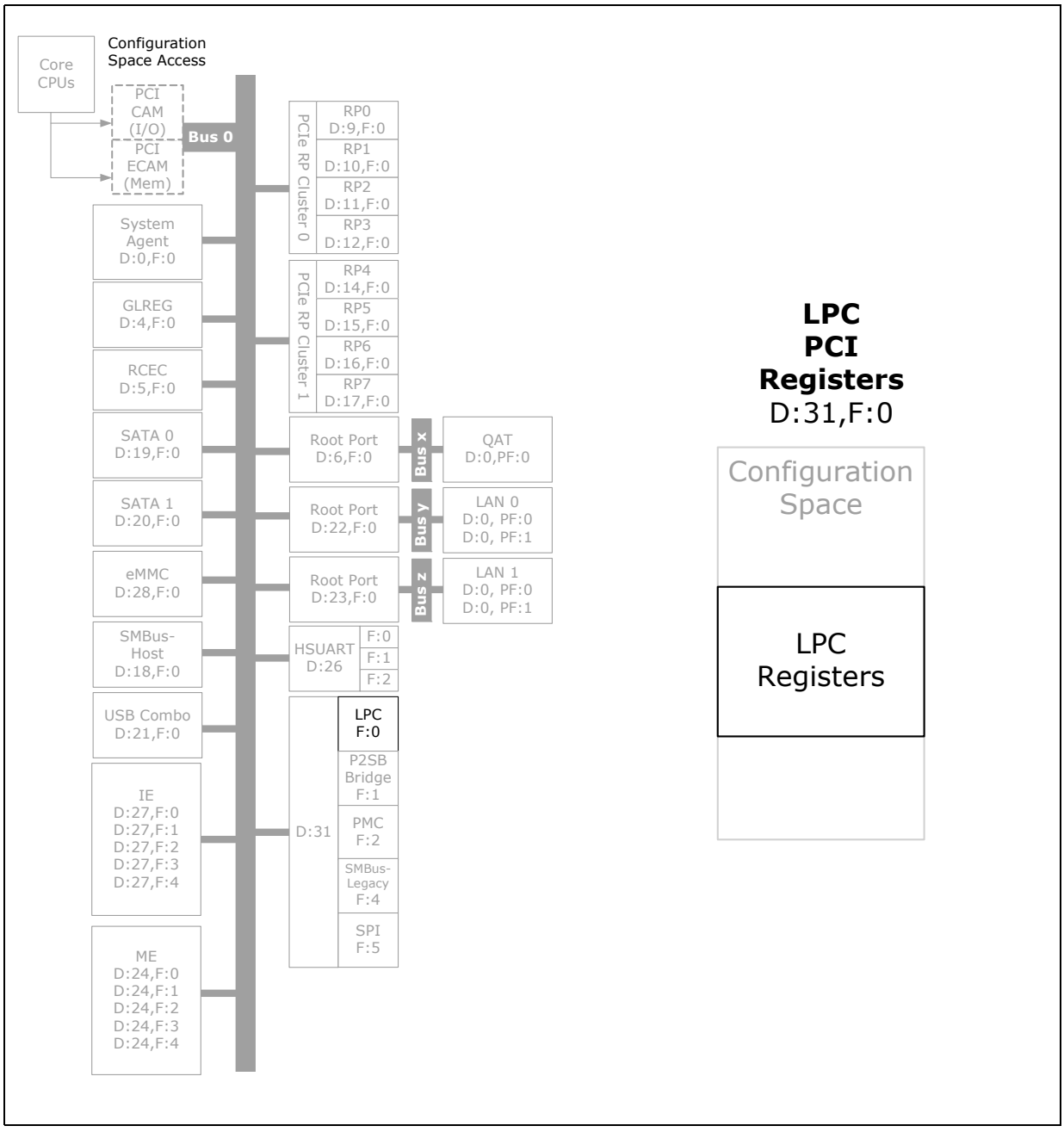
| Symbol | Minimum | Maximum | Units | Description |
|---|---|---|---|---|
| t90 | 29 | 30 | LPC CLK | No activity to idle detect |
| t91 | 1 | 1 | LPC CLK | Idle detect to LPC_CLKRUN_N high |
| t92 | 2 | 3 | LPC CLK | LPC_CLKRUN_N high to LPC_CLKRUN_N (no stop) [if low from external source]<br>***Note:*** The maximum can be 5 if LPC_CLKRUN_N low (no stop) due to an internal source |
| t93 | 6 | | LPC CLK | LPC_CLKRUN_N high (stop) |
| t94 | 1 | | LPC CLK | LPC_CLKRUN_N low (restart) |

*Note:* Timing t94 only applies to external devices asserting LPC_CLKRUN_N. If an internal source is active and requests for LPC CLK to be restarted, the SoC will assert LPC_CLKRUN_N. This is because clock synchronization is not required.

## 20.4 Register Map

Figure 20-3 shows the associated registers from a system software viewpoint.

**Figure 20-3. Register Map**



### 20.4.1 PCI Configuration and Capabilities

**Table 20-4. LPC Register Map - Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D31F:F0 (hexadecimal) | Register Short Name | Description |
|---|---|---|
| 00 | ID | Identifiers |
| 04 | CMD | Command |
| 06 | STS | Device Status |
| 08 | RID | Revision ID |
| 09 | CC | Class Code |
| 0D | PLT | Primary Latency Timer |
| 0E | HTYPE | Header Type |
| 2C | SS | Subsystem Identifiers |
| 34 | CAPP | Capability List Pointer |
| 64 | SCNT | Serial IRQ Control |
| 80 | IOD | I/O Decode Ranges |
| 82 | IOE | I/O Enables |
| 84 | LGIR1 | LPC Generic I/O Range #1 |
| 88 | LGIR2 | LPC Generic I/O Range #2 |
| 8C | LGIR3 | LPC I/O Range #3 |
| 90 | LGIR4 | LPC I/O Range #4 |
| 94 | ULKMC | USB Legacy Keyboard/Mouse Control |
| 98 | LGMR | LPC Generic Memory Range |
| 9C | RSVD | Reserved |
| D0 | FS1 | FWH ID Select #1 |
| D4 | FS2 | FWH ID Select #2 |
| D8 | BDE | BIOS Decode Enable |
| DC | BC | BIOS Control |
| E0 | PCCTL | PCI Clock Control |

## 20.4.2 Fixed I/O Address Transactions

This subsection lists the claimable fixed I/O addresses by the LPC controller on the primary interface.

**Table 20-5. LPC Claimable Fixed I/O Range as Target on Primary Interface (Sheet 1 of 2)**

<table>
<tr><th>#</th><th>Address Decode Range</th><th>Allocatable Size (bytes)</th><th>Enable B0:D31:F0 + Offset</th><th>Description</th></tr>
<tr><td>1</td><td>200h – 207h</td><td>8</td><td>0x82h[9]</td><td>Gameport High</td></tr>
<tr><td>2</td><td>208h – 20Fh</td><td>8</td><td>0x82h[8]</td><td>Gameport Low</td></tr>
<tr><td>3</td><td>B0:D31:F0+0x80h[12]<br><table><tr><th>Bits</th><th>Decode Range</th></tr><tr><td>0</td><td>3F0h - 3F5h, 3F7h (primary)</td></tr><tr><td>1</td><td>370h - 375h, 377h (secondary)</td></tr></table>2 ranges in 64k I/O space</td><td>8</td><td>0x82h[3]</td><td>Floppy Disk Drive Port</td></tr>
<tr><td>4</td><td>B0:D31:F0+0x80h[9:8]<br><table><tr><th>Bits</th><th>Decode Range</th></tr><tr><td>00</td><td>378h - 37Fh and 778h - 77Fh</td></tr><tr><td>01</td><td>278h - 27Fh (port 279h is read only) and 678h - 67Fh</td></tr><tr><td>10</td><td>3BCh - 3BEh and 7BCh - 7BEh</td></tr><tr><td>11</td><td>Reserved</td></tr></table>3 ranges in 64k I/O space</td><td>8</td><td>0x82h[2]</td><td>LPC Parallel Port</td></tr>
<tr><td>5</td><td>B0:D31:F0+0x80h[6:4]<br><table><tr><th>Bits</th><th>Decode Range</th></tr><tr><td>000</td><td>3F8h - 3FFh (COM 1)</td></tr><tr><td>001</td><td>2F8h - 2FFh (COM 2)</td></tr><tr><td>010</td><td>220h - 227h</td></tr><tr><td>011</td><td>228h - 22Fh</td></tr><tr><td>100</td><td>238h - 23Fh</td></tr><tr><td>101</td><td>2E8h - 2EFh (COM 4)</td></tr><tr><td>110</td><td>338h - 33Fh</td></tr><tr><td>111</td><td>3E8h - 3EFh (COM 3)</td></tr></table>8 ranges in 64k I/O space</td><td>8</td><td>0x82h[1]</td><td>ComB Port</td></tr>
</table>

## Table 20-5. LPC Claimable Fixed I/O Range as Target on Primary Interface (Sheet 2 of 2)

<table>
<tr><th>#</th><th>Address Decode Range</th><th>Allocatable Size (bytes)</th><th>Enable B0:D31:F0 + Offset</th><th>Description</th></tr>
<tr><td>6</td><td>B0:D31:F0+0x80h[2:0]<br><table><tr><th>Bits</th><th>Decode Range</th></tr><tr><td>000</td><td>3F8h - 3FFh (COM 1)</td></tr><tr><td>001</td><td>2F8h - 2FFh (COM 2)</td></tr><tr><td>010</td><td>220h - 227h</td></tr><tr><td>011</td><td>228h - 22Fh</td></tr><tr><td>100</td><td>238h - 23Fh</td></tr><tr><td>101</td><td>2E8h - 2EFh (COM 4)</td></tr><tr><td>110</td><td>338h - 33Fh</td></tr><tr><td>111</td><td>3E8h - 3EFh (COM 3)</td></tr></table>8 ranges in 64k I/O space</td><td>8</td><td>0x82h[0]</td><td>ComA Port</td></tr>
</table>

## 20.4.3 Configurable I/O Address Transactions

This subsection lists the claimable configurable I/O addresses by the LPC controller on the primary interface.

**Table 20-6. LPC Claimable Configurable I/O Range as Target on Primary Interface**

| # | Address Decode Range | Allocatable Size (bytes) | Enable B0:D31:F0 + Offset | Description |
|---|---|---|---|---|
| 1 | 4 to 256 bytes I/O block size anywhere in the 64k I/O space | 4 – 256 | 0x84h[0] | LPC Generic I/O Range #1 |
| 2 | | | 0x88h[0] | LPC Generic I/O Range #2 |
| 3 | | | 0x8Ch[0] | LPC Generic I/O Range #3 |
| 4 | | | 0x90h[0] | LPC Generic I/O Range #4 |

## 20.4.4 Memory and BIOS Address Transactions

This subsection lists the claimable memory and BIOS addresses by the LPC controller on the primary interface.

### Table 20-7. LPC Claimable Memory Mapped Range as Target on Primary Interface

| # | Address Decode Range | Allocatable Size (bytes) | Enable B0:D31:F0 + Offset | Claiming Agent | Description |
|---|---|---|---|---|---|
| 1 | 64 KB memory block anywhere in the 4 GB memory space | 64 KB | 0x98h[0] | LPC or Serial Peripheral Interface (SPI) primary. The source decode to determine the claiming agent using the BBS and LPC SoC hard (pin) straps.<br><br>{BBS, LPC decode}<br>00 → SPI<br>01 → SPI<br>10 → LPC[1] | LPC Generic Memory Range |
| 2[1] | 000E_0000h – 000E_FFFFh | 64 KB | 0xD8h[6] | | BIOS Legacy E segment |
| 3[1] | 000F_0000h – 000F_FFFFh | 64 KB | 0xD8h[7] | | BIOS Legacy F segment |
| 4 | FFC0_0000h – FFC7_FFFFh | 512 KB | 0xD8h[8] | | BIOS C0-C7 Data space |
| 5 | FF80_0000h – FF87_FFFFh | 512 KB | 0xD8h[8] | | BIOS C0-C7 Feature space |
| 6 | FFC8_0000h – FFCF_FFFFh | 512 KB | 0xD8h[9] | | BIOS C8-CF Data space |
| 7 | FF88_0000h – FF8F_FFFFh | 512 KB | 0xD8h[9] | | BIOS C8-CF Feature space |
| 8 | FFD0_0000h – FFD7_FFFFh | 512 KB | 0xD8h[10] | | BIOS D0-D7 Data space |
| 9 | FF90_0000h – FF97_FFFFh | 512 KB | 0xD8h[10] | | BIOS D0-D7 Feature space |
| 10 | FFD8_0000h – FFDF_FFFFh | 512 KB | 0xD8h[11] | | BIOS D8-DF Data space |
| 11 | FF98_0000h – FF9F_FFFFh | 512 KB | 0xD8h[11] | | BIOS D8-DF Feature space |
| 12 | FFE0_0000h – FFE7_FFFFh | 512 KB | 0xD8h[12] | | BIOS E0-E7 Data space |
| 13 | FFA0_0000h – FFA7_FFFFh | 512 KB | 0xD8h[12] | | BIOS E0-E7 Feature space |
| 14 | FFE8_0000h – FFEF_FFFFh | 512 KB | 0xD8h[13] | | BIOS E8-EF Data space |
| 15 | FFA8_0000h – FFAF_FFFFh | 512 KB | 0xD8h[13] | | BIOS E8-EF Feature space |
| 16 | FFF0_0000h – FFF7_FFFFh | 512 KB | 0xD8h[14] | | BIOS F0-F7 Data space |
| 17 | FFB0_0000h – FFB7_FFFFh | 512 KB | 0xD8h[14] | | BIOS F0-F7 Feature space |
| 18 | FFF8_0000h – FFFF_FFFFh | 512 KB | Always enabled | | BIOS F8-FF Data space. Swappable using Top Swap soft strap. |
| 19 | FFB8_0000h – FFBF_FFFFh | 512 KB | Always enabled | | BIOS F8-FF Feature space. Not Swappable using Top Swap soft strap. Chapter 4, "Strapping and Configuration" |
| 20 | FF70_0000h – FF7F_FFFFh | 1 MB | 0xD8h[3] | | BIOS 70-7F Data space |
| 21 | FF30_0000h – FF3F_FFFFh | 1 MB | 0xD8h[3] | | BIOS 70-7F Feature space |
| 22 | FF60_0000h – FF6F_FFFFh | 1 MB | 0xD8h[2] | | BIOS 60-6F Data space |
| 23 | FF20_0000h – FF2F_FFFFh | 1 MB | 0xD8h[2] | | BIOS 60-6F Feature space |
| 24 | FF50_0000h – FF5F_FFFFh | 1 MB | 0xD8h[1] | | BIOS 50-5F Data space |
| 25 | FF10_0000h – FF1F_FFFFh | 1 MB | 0xD8h[1] | LPC or SPI primary. The source decode to determine the claiming agent using the boot BIOS and LPC SoC hard (pin) straps. | BIOS 50-5F Feature space |
| 26 | FF40_0000h – FF4F_FFFFh | 1 MB | 0xD8h[0] | | BIOS 40-4F Data space |
| 27 | FF00_0000h – FF0F_FFFFh | 1 MB | 0xD8h[0] | | BIOS 40-4F Feature space |

1. Intel is not validating legacy support.

### 20.4.4.1 BIOS Write Protect and Top Swap Range

Table 20-8, "LPC Claimable Configurable I/O Range as Target on Primary Interface" lists the BIOS Decode Address range that are subjected to the Write Protect Disable, Enable InSMM.STS (EISS) and Top Swap configuration.

**Table 20-8. LPC Claimable Configurable I/O Range as Target on Primary Interface**

| BIOS Decode Address Range | WPD | EISS | Writes | Top Swap |
|---|---|---|---|---|
| All ranges: Feature Space | X | X | Allowed | No |
| Other BIOS decode ranges (Data Space) & Legacy E/F Segments | 0 | X | Blocked | Only apply to EF8 (FFF80000h - FFFFFFFFh) |
| | 1 | 0 | Allowed | |
| ***Note:*** Intel is not validating legacy support. | 1 | 1 | Allowed if InSMM.STS = 1 | |

### 20.4.4.2 BIOS Boot Decode Claiming Agent

Table 20-9, "BIOS Boot Decode Claiming Agent" lists the claiming agents for the BIOS cycles to access the Boot codes and the respective BIOS Decode Range Register location. The Boot BIOS Pin Strap goes out to 2 locations, namely SPI and LPC. The BBS register for these registers are programmable to override the SoC hard (pin) strap value. BIOS needs to ensure that the BBS values on both locations (SPI and LPC) are set the same (match).

**Table 20-9. BIOS Boot Decode Claiming Agent**

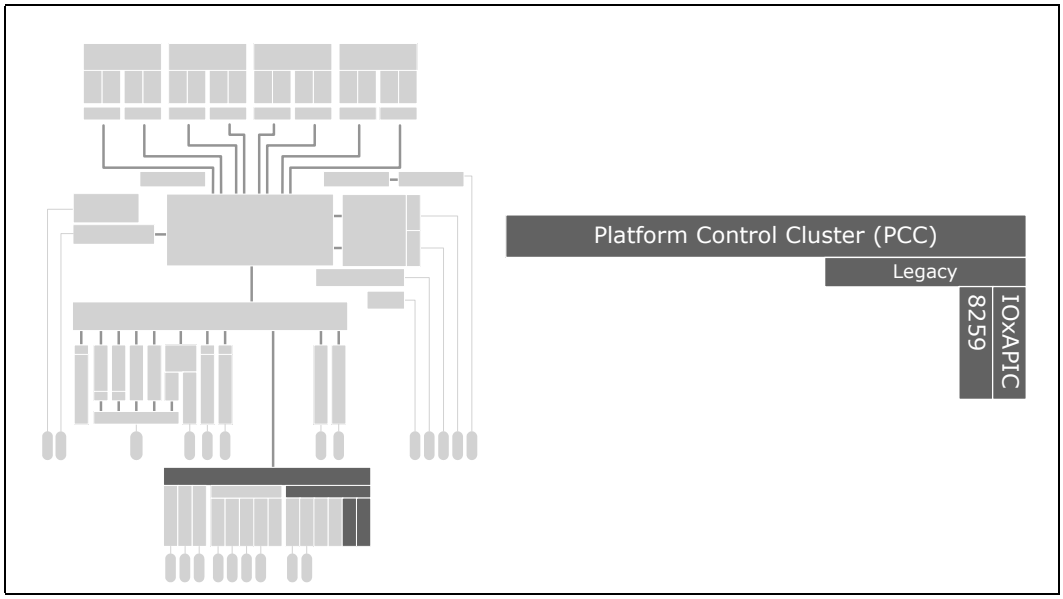
| Pin BR58 Boot BIOS Strap (BBS) | Pin BL52 LPC Decode Select | BIOS Boot Location | BIOS Boot Decode Range Register Location |
|---|---|---|---|
| 0 | 0 | SPI | SPI PCI Configuration Register.<br>Program SPI PCI-configuration (D31:F5) BIOS Decode Range Configuration Registers. SPI uses its own BIOS Decode<br>Range to determine the BIOS Boot Location |
| 0 | 1 | SPI | |
| 1 | 0 | LPC | LPC PCI Configuration Register.<br>Program LPC PCI-configuration (D31:F0) BIOS Decode Range Configuration Registers. LPC uses its own BIOS Decode Range to determine the BIOS Boot Location. |

§ §

# 21 PIC and I/O APIC Controllers

This chapter describes the Programmable Interrupt Controller (PIC). The design is based on the legacy 8259 PIC device. This chapter also describes the I/O Advanced Programmable Interrupt Controller (IOxAPIC).

**Figure 21-1. What is Covered in This Chapter**



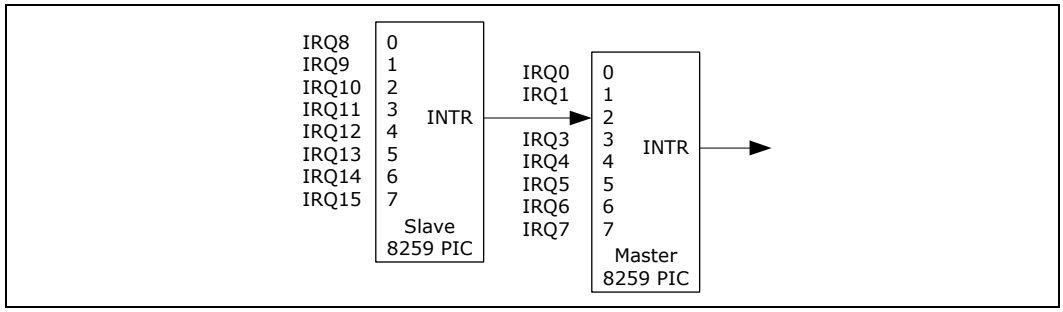
## 21.1 Signal Descriptions

There are no signal pins or signal names associated with this chapter.

## 21.2 Programmable Interrupt Controller (PIC)

The design is based on the legacy 8259 PIC device. The slave controller is cascaded onto the master controller through master controller interrupt input 2. This means there are 15 possible Interrupt Requests (IRQ) routed internally to the SoC PIC, see Figure 21-2. The interrupt routing is described in Chapter 8, "Interrupts and Events".

**Figure 21-2. Master and Slave PIC**




### 21.2.1 Special Internal Circuitry

Interrupts can be individually-programmed to be edge-triggered or level-sensitive, except for IRQ0, IRQ1, IRQ2 and IRQ8 which always default to edge-triggered.

Active-low interrupt sources, such as the PIRQ#, are internally inverted before being sent to the PIC. In the following descriptions of the 8259, the interrupt levels are in reference to the signals at the internal interface of the 8259, after the required inversions have occurred. Therefore, the term "high" indicates "active", which means "low" on an originating PIRQ#.

### 21.2.2 Generating Interrupts

The PIC interrupt sequence involves 3 bits, from the IRR, ISR, and IMR, for each interrupt level. These bits are used to determine the interrupt vector returned and the status of any other pending interrupts. Table 21-1 defines the IRR, ISR, and IMR.

**Table 21-1. Interrupt Status Registers**

| Bit | Description |
|---|---|
| IRR | **Interrupt Request Register:** This bit is set on a low-to-high transition of the interrupt line in the edge mode and by an active high level in the level mode. This bit is set whether or not the interrupt is masked. However, a masked interrupt will not generate the internal INTR signal. |
| ISR | **Interrupt Service Register:** This bit is set, and the corresponding IRR bit cleared, when an interrupt acknowledge cycle is seen, and the vector returned is for that interrupt. |
| IMR | **Interrupt Mask Register:** This bit determines whether an interrupt is masked. Masked interrupts do not generate the internal INTR signal. |

### 21.2.3 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated into a interrupt acknowledge cycle to the SoC. The PIC translates this command into two internal INTA# pulses expected by the 8259 controllers. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based on the five Interrupt Vector Base Address (IVBA) bits of the Master/Slave Initialization Command Word 2 (MICW2 or SICW2), combined with the three Interrupt Request Level (IRL) bits of the MICW2 or SICW2 which represent the interrupt within that controller.

Table 21-2 applies to the Master 8259 controller and to the Slave 8259 controller.

**Table 21-2. Content of Interrupt Vector Byte**

| Master, Slave Interrupt to be serviced by the Interrupt Acknowledge cycle | Master/Slave ICW2 Bits [7:3] generated by the controller | Master/Slave ICW2 Bits [2:0] generated by the controller |
|---|---|---|
| IRQ7,15 | Interrupt Vector Base Address (IVBA) in the Interrupt Vector Table for the interrupt routine | 111 |
| IRQ6,14 | | 110 |
| IRQ5,13 | | 101 |
| IRQ4,12 | | 100 |
| IRQ3,11 | | 011 |
| IRQ2,10 | | 010 |
| IRQ1,9 | | 001 |
| IRQ0,8 | | 000 |

### 21.2.4 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in the edge mode or seen high in the level mode, setting the corresponding IRR bit.
2. The PIC sends the internal INTR signal to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an internal interrupt acknowledge cycle.
4. When observing the special cycle, the SoC converts it into the two cycles that the internal 8259 pair need to responds. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# signal of the cascaded interrupt controllers.
5. When receiving the first internally-generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse, a slave identification code is broadcast by the master to the slave on a private, internal 3-bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second, internal INTA# pulse.
6. When receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC indicates a spurious interrupt by returning interrupt vector 7 from the master controller.
7. This completes the interrupt cycle. In Automatic End of Interrupt (AEOI) mode, the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

## 21.2.5 Initialization Command Words (ICWx)

Before an operation begins, each 8259 must be initialized. In the SoC, this is a four-byte sequence. The four Initialization Command Words (ICW) are referred by their acronyms: ICW1, ICW2, ICW3, and ICW4.

The base address for each 8259 initialization command word is a fixed location in Host I/O Space address 20h for the master controller and address A0h for the slave controller.

### 21.2.5.1 ICW1

A write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the PIC expects three more byte writes to 21h for the master controller or A1h for the slave controller to complete the ICW sequence.

A write to ICW1 starts the initialization sequence during which the following automatically occurs:

1. Following initialization, an Interrupt Request (IRQ) input must make a low-to-high transition to generate an interrupt.
2. The Interrupt Mask Register is cleared.
3. IRQ7 input is assigned priority 7.
4. The slave mode address is set to 7.
5. Special mask mode is cleared, and the status read is set to IRR.

### 21.2.5.2 ICW2

The second write in the sequence (ICW2) is programmed to provide bits [7:3] of the interrupt vector that are released during an interrupt acknowledge. A different base is selected for each interrupt controller. See Table 21-2, “Content of Interrupt Vector Byte” on page 640.

### 21.2.5.3 ICW3

The third write in the sequence (ICW3) has a different meaning for each controller.

- For the master controller, ICW3 indicates which of the master’s IRQ input lines cascades the IRQ[15:8] of the slave controller. Within the SoC, IRQ2 is used. Therefore, the Cascaded Controller Connection (CCC) indication, bit 2 of the MICW3 is set to a 1 and the other bits are set to 0s.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3, and if it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

### 21.2.5.4 ICW4

The final write in the sequence (ICW4) must be programmed for the master controller and for the slave controller. At least the Microprocessor Mode (MM) indication, bit 0 of the ICW4, must be set to a 1 to indicate the controllers are operating in an Intel® architecture system.

### 21.2.6 Operation Command Words (OCW)

These command words reprogram the interrupt controller to operate in various interrupt modes:

- OCW1 masks and unmasks the interrupt lines.
- OCW2 controls the rotation of interrupt priorities when in rotating priority mode and controls the EOI function.
- OCW3 sets up the ISR/IRR reads, enables/disables the Special Mask Mode (SMM), and enables/disables the polled interrupt mode.

### 21.2.7 Fully-Nested Mode

In this mode, interrupt requests are ordered in priority from 0 through 7, with 0 being the highest. When an interrupt is acknowledged, the highest priority request is determined and its vector placed on the bus. Additionally, the ISR for the interrupt is set. This ISR bit remains set until: the processor issues an End of Interrupt (EOI) command immediately before returning from the service routine or if in Automatic End of Interrupt (AEOI) mode, on the trailing edge of the second, internal INTA# signal. While the ISR bit is set, all further interrupts of the same or lower priority are inhibited, while higher levels generate another interrupt. Interrupt priorities are changed in the rotating priority mode.

### 21.2.8 Special Fully-Nested Mode

This mode is used in the case of a system where cascading is used, and the priority has to be conserved within each slave. In this case, the special fully-nested mode is programmed to the master controller. This mode is similar to the fully-nested mode with the following exceptions:

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master priority logic, and further interrupt requests from higher priority interrupts within the slave are recognized by the master and initiate the interrupts to the processor. In the normal-nested mode, a slave is masked out when its request is in service.
- When exiting the interrupt service routine, the software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI is also sent to the master.

### 21.2.9 Automatic Rotation Mode (Equal Priority Devices)

In some applications, there are a number of interrupting devices of equal priority. The automatic rotation mode provides for a sequential eight-way rotation. In this mode, a device receives the lowest priority after being serviced. In the worst case, a device requesting an interrupt has to wait until each of the seven other devices are serviced at most once.

Two ways to accomplish automatic rotation using the three-bit Rotate and EOI Codes (REOI) bits [7:5] of the OCW2 are:

101 = Rotation on the Non-Specific EOI command
100 = Rotation in the Automatic EOI mode

### 21.2.10 Specific Rotation Mode (Specific Priority)

The software changes the interrupt priorities by programming the bottom priority. For example, if IRQ5 is programmed as the bottom-priority device, then IRQ6 is the highest-priority device. This is accomplished by using the three-bit Rotate and EOI Codes (REOI) field, bits [7:5] of the OCW2:

110 = Set Priority command

along with the three-bit Interrupt Level Select (L2, L1, L0) (ILSLT) field, bits [2:0] of the OCW2 to indicate the binary priority level code of the bottom priority device.

In this mode, the internal status is updated by the software control during OCW2. However, it is independent of the EOI command. Priority changes are executed during an EOI command by using REOI set to:

111 = Rotate on Specific EOI command

along with the three-bit ILSLT to indicate the binary priority level code of the bottom priority device.

### 21.2.11 Poll Mode

The poll mode conserves space in the interrupt vector table. Multiple interrupts that are serviced by one interrupt service routine do not need separate vectors if the service routine uses the Poll command. The poll mode is also used to expand the number of interrupts. The polling interrupt service routine calls the appropriate service routine instead of providing the interrupt vectors in the vector table. In this mode, the internal INTR output signal is not used and the processor internal interrupt enable flip-flop is reset, disabling its interrupt input. Service to the devices is achieved by the software using a Poll command.

The Poll command is issued by setting the Poll Mode Command (PMC), bit 2 of the OCW3. The PIC treats its next Host I/O Read as an interrupt acknowledge, sets the appropriate ISR bit if there is a request, and reads the priority level. Interrupts are frozen from the OCW3 write to the Host I/O Read. The byte returned during the Host I/O Read contains a 1 in bit 7 if there is an interrupt, and the binary code of the highest-priority level is indicated by bits [2:0].

### 21.2.12 Edge- and Level-Triggered Mode

In Industry Standard Architecture (ISA) systems, this mode is programmed using the Level-Triggered Interrupt Mode (LITM), bit 3 of the ICW1, which sets level or edge for the entire controller. In the SoC, this bit is disabled and a register for edge- and level-triggered mode selection is provided per interrupt input. These make up the Master Edge/Level Control (ELCR1) and Slave Edge/Level Control (ELCR2) registers.

If the ELCR bit is 0, an interrupt request is recognized by a low-to-high transition on the corresponding IRQ input. The IRQ input remains high without generating another interrupt. If the ELCR bit is 1, an interrupt request is recognized by a high level on the corresponding IRQ input, and an edge detection is not needed. The interrupt request must be removed before the EOI command is issued to prevent a second interrupt from occurring.

In both the edge- and level-triggered modes, the IRQ inputs must remain active until after the falling edge of the first internal INTA# signal. If the IRQ input goes inactive before this time, a default IRQ7 vector is returned indicating a spurious interrupt.

## 21.2.13 End of Interrupt (EOI) Operations

An EOI occurs in one of two fashions:

- Normal EOI - by a command word write issued to the PIC before returning from a service routine, the EOI command
- Automatic EOI - when the Automatic End of Interrupt (AEOI), bit 1 of ICW4, is set to 1.

### 21.2.13.1 Normal End of Interrupt (EOI) Mode

In normal EOI mode, the software writes an EOI command before leaving the interrupt service routine to mark the interrupt as completed. The two forms of EOI commands are: Specific and Non-Specific.

When a Non-Specific EOI command is issued, the PIC clears the highest ISR bit of those that are set to 1. A Non-Specific EOI is the normal mode of operation of the PIC within the SoC, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in the modes that preserve the fully-nested structure, the software determines which ISR bit to clear by issuing a specific EOI.

An ISR bit that is masked is not cleared by a non-specific EOI if the PIC is in the special mask mode. An EOI command must be issued for both the master and slave controller.

### 21.2.13.2 Automatic End of Interrupt (AEOI) Mode

In this mode, the PIC automatically performs a Non-Specific EOI operation at the trailing edge of the last interrupt acknowledge pulse. From a system standpoint, this mode is used only when a nested multi-level interrupt structure is not required within a single PIC. The AEOI mode is only used in the master controller and not the slave controller.

*Note:* Both the master and slave PICs have an AEOI bit in MICW4 and SICW4 respectively. Only the AEOI bit of MICW4 is set by the software. The AEOI bit of SICW4 is not set by the software.

## 21.2.14 Masking Interrupts

### 21.2.14.1 Masking on an Individual Interrupt Request

Each interrupt request is masked individually by the Interrupt Mask Register (IMR). This register is programmed through OCW1. Each bit in the IMR masks one interrupt channel. Masking IRQ2 on the master controller masks all requests for service from the slave controller.

### 21.2.14.2 Special Mask Mode

Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control. For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.

The Special Mask Mode enables all interrupts not masked by a bit set in the Mask Register. Normally, when an interrupt service routine acknowledges an interrupt without issuing an EOI to clear the ISR bit, the interrupt controller inhibits all lower priority requests. In the special mask mode, any interrupts are selectively enabled by loading the mask register with the appropriate pattern.

The Special Mask Mode is set when the Enable Special Mask Mode (ESMM), bit 5 of OCW3, and the Special Mask Mode (SMM), bit 6 of OCW3, are programmed to 1. Special Mask Mode is cleared when ESMM = 1 and SMM = 0. The SMM bit has no meaning when ESSM is 0.

## 21.3 I/O Advanced Programmable Interrupt Controller (IOxAPIC)

- Up to 120 interrupt requests (IRQ0 through IRQ119). Default-set to 24.
- Edge- or level-trigger mode per interrupt
- Active-low or -high polarity per interrupt
- Uses an MSI-like memory write cycle to send interrupts for local (processor) APICs
- Established APIC programming model

### 21.3.1 Interrupt Redirection Table

The interrupt Redirection Table of the SoC IOxAPIC provides 120 64-bit I/O Interrupt Redirection Table Entry (RTE) registers. Each RTE register is dedicated to one of the 120 internal, Interrupt Request (IRQ) input signals. For information about interrupts routed to the IOxAPIC see Chapter 8, "Interrupts and Events".

The information in each RTE is used to translate the interrupt manifestation on the corresponding IOxAPIC Interrupt Request input signal into a message to the local APICs.

The Maximum Redirection Entries (MRE) is default-set to indicate 24 (decimal) interrupts, IRQ[23:0]. BIOS can change this maximum number through the Read-Write-Once (RWO), eight-bit MRE field of the Version (VS) register. MRE contains the RTE number (0 being the lowest entry) of the highest entry in the Redirection Table.

BIOS must write to the MRE field after the SoC asserts the active-low PMU_PLTRST_N platform reset signal pin in order to lock-down the RWO value. This allows BIOS to utilize some of the entries for its own purpose and thus advertising fewer IOxAPIC Redirection Entries to OS. BIOS may program the eight-bit MRE field with a value up to 78h (maximum of 120 RTE registers).

Unlike IRQ pins of the 8259, the notion of interrupt priority is completely unrelated to the position of the physical interrupt input signal on the IOxAPIC. Instead, the software determines the vector (and therefore the priority) for each corresponding interrupt input signal. For each interrupt signal, the operating system also specifies the signal polarity (low active or high active), whether the interrupt is signaled as edges or levels, as well as the destination and delivery mode of the interrupt.

The information in the Redirection Table translates the corresponding interrupt pin information into an inter-APIC message.

### 21.3.2 Accessing the Redirection Table

A 32-bit Identification (ID) register, 32-bit Version (VS) register, and 120 64-bit RTE registers are accessible in Host Memory Space through an Index (IDX) - Window (WDW) mechanism. The mechanism itself can be relocated within Host Memory Space. See Table 21-6, "IOxAPIC Register Access Window in Host Memory Space" on page 652 and Table 21-7, "IOxAPIC Registers" on page 652.

When accessing the indexed registers, accesses must be done as 32-bit Double-Words or else unspecified behavior results. Software must not attempt to write to reserved registers. Some reserved registers may return non-zero values when read.

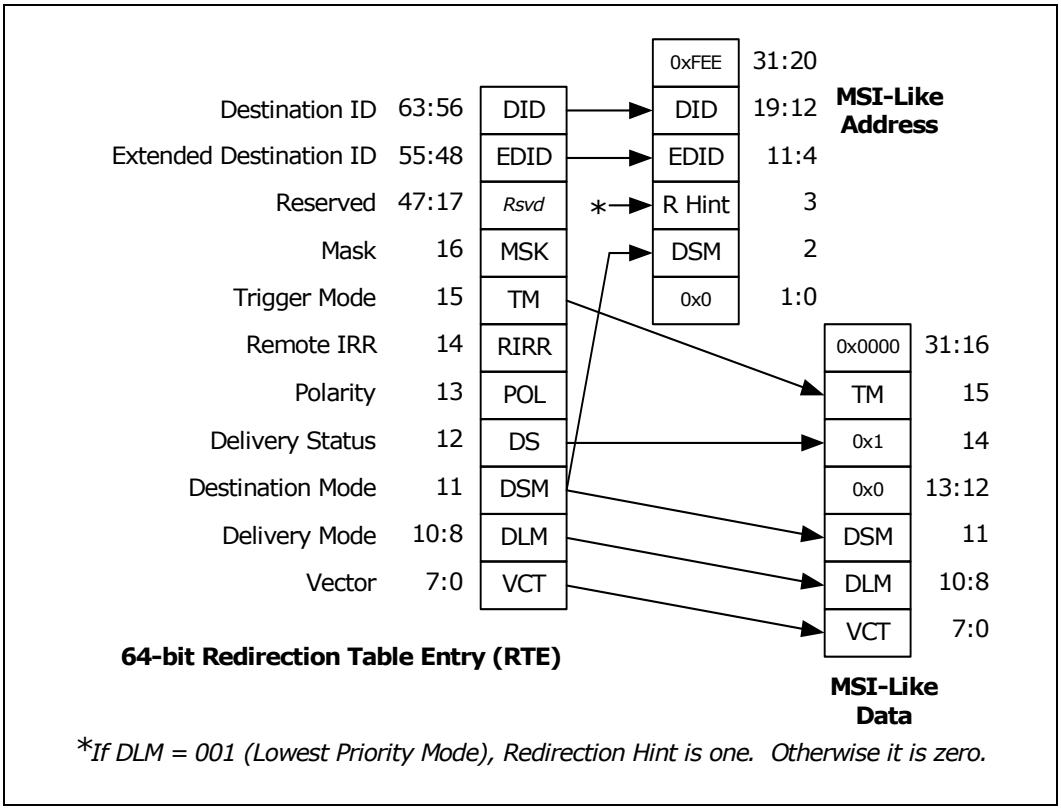
### 21.3.3 IOxAPIC Interrupts to the Local APICs

The VT-d architecture extension requires Interrupt Messages to go through the similar Address Remapping as any other memory requests. This is to allow domain isolation for interrupts such that a device assigned in one domain is not allowed to generate interrupts to another domain.

The Address Remapping for VT-d is based on the Bus:Device:Function field associated with the requests. Hence, it is required for the internal IOxAPIC to initiate the Interrupt Messages using a unique Bus:Device:Function.

The SoC Primary-to-Sideband (P2SB) Bridge PCI device (B:0, D:31, F:1) provides this VT-d capability and issues the MSI-like, IOxAPIC Interrupt Message to the processor local APICs on behalf of the IOxAPIC as an upstream 32-bit Memory Write-Cycle Transaction. The P2SB Bridge B/D/F is used in the Requester ID when the IOxAPIC initiates sending Interrupt Messages to the CPU. The Host Memory-Space address and data values are derived from the Relocation table Entry (RTE) for the interrupt and follow the format shown in Figure 21-3 on page 647.

**Figure 21-3. MSI-Like IOxAPIC Interrupt Message for the Local APICs**



The P2SB Bridge B/D/F is also used as the Completer ID in response to reads targeting the IOxAPICs Memory-Mapped I/O (MMIO) registers. See Section 21.3.2, "Accessing the Redirection Table" on page 646.

*Note:* While the P2SB Bridge is the designated PCI device to act on behalf of the IOxAPIC Interrupt Message Writes and Reads of its MMIO registers, there is a register that determines the B/D/F values for the IOxAPIC's surrogate PCI device. The register is the IOxAPIC Bus:Device:Function (IBDF) and is located in Host Configuration Space of the P2SB Bridge, offset 6Ch. The default for IBDF is B:0, D:31, F:0 (LPC) and must be changed to B:0, D:31, F:1 (P2SB Bridge).

The Host Memory Space address of the IOxAPIC Interrupt Message is formed as shown in Table 21-3. The data is shown in Table 21-4.

### Table 21-3. IOxAPIC Interrupt Message Address

| Address Bit | Short Description | Description |
|---|---|---|
| 31:20 | FEEh | Hardwired as FEEh |
| 19:12 | Destination ID | Destination ID of the local APIC - from DID field of the interrupt RTE |
| 11:4 | Extended Destination ID | Extended destination ID of the local APIC - from EDID field of the interrupt RTE |
| 3 | Redirection Hint | Redirection Hint - '1' If the DLM field = "Lowest Priority" (001) for the interrupt RTE. Otherwise '0' |
| 2 | Destination Mode | Used by the Local APIC to determine whether it is the destination of the Interrupt Message |
| 1:0 | 00 | Hardwired as 00 - Interrupt Message data is written as a Double Word (32 bits) |

### Table 21-4. IOxAPIC Interrupt Message Data

| Data Bit | Short Description | Description |
|---|---|---|
| 31:16 | 0000h | Hardwired as 0000h |
| 15 | Trigger Mode | From TM field of the RTE - '0' indicates the interrupt is edge sensitive. '1' indicates the interrupt is level sensitive. |
| 14 | Delivery Status | Is always '1' indicating an "Assert Message" cycle |
| 13:12 | 00 | Hardwired as 00 |
| 11 | Destination Mode | From DSM field of the RTE - Used by the Local APIC to determine whether it is the destination of the message |
| 10:8 | Delivery Mode | From DLM field of the RTE - How the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. The supported modes are described in Section 21.3.4, "Interrupt Delivery Modes" on page 649. |
| 7:0 | Vector | From VCT field of the RTE - Interrupt Vector for this interrupt. Values range between 10h and Feh |

### 21.3.4 Interrupt Delivery Modes

The IOxAPIC supports these Interrupt Delivery Modes (DLM) to the local APICs:

- 000 = Fixed. Deliver the signal on the INTR (internal) signal of all processor cores listed in the Destination field of the RTE. The Trigger Mode field of the RTE can be edge or level.
- 001 = Lowest Priority. Deliver the signal on the INTR (internal) signal of the processor core that is executing at the lowest priority among all the processors listed in the specified Destination field of the RTE. Trigger Mode can be edge or level.
- 111 = ExtINT. Deliver the signal to the INTR signal of all processor cores listed in the Destination field as an interrupt that originated in the 8259 interrupt controller. The INTA cycle that corresponds to this ExtINT delivery is routed to the 8259 controller which is expected to supply the vector. Requires the interrupt to be programmed as edge triggered.

The IOxAPIC does not support the NMI, INIT, or SMI Delivery Modes. However, the SoC does send the Delivery Mode (DLM) value programmed in the Redirection Table as part of the Interrupt Message.

*Note:* Software is required to take precaution as the SoC does not internally route the NMI/INIT/SMI events to the IOxAPIC, nor attempting to resolve any dependency with the NMI/INIT/SMI delivery as described above even though the DLM field is programmed as such. If needed, this must be addressed outside of the SoC.

### 21.3.5 Determining Safe Message Delivery Time

When the IOxAPIC sends an Interrupt Message, it first ensures that all buffers that may contain main-memory-directed posted writes from legacy devices have been flushed to the point that when the interrupt is delivered to the CPU, any write data that needs to be processed by the Interrupt Service Software is available in the memory complex. When the legacy-device buffers are empty, the IOxAPIC delivers the interrupt that initiated the flush.

### 21.3.6 End Of Interrupt (EOI) Register

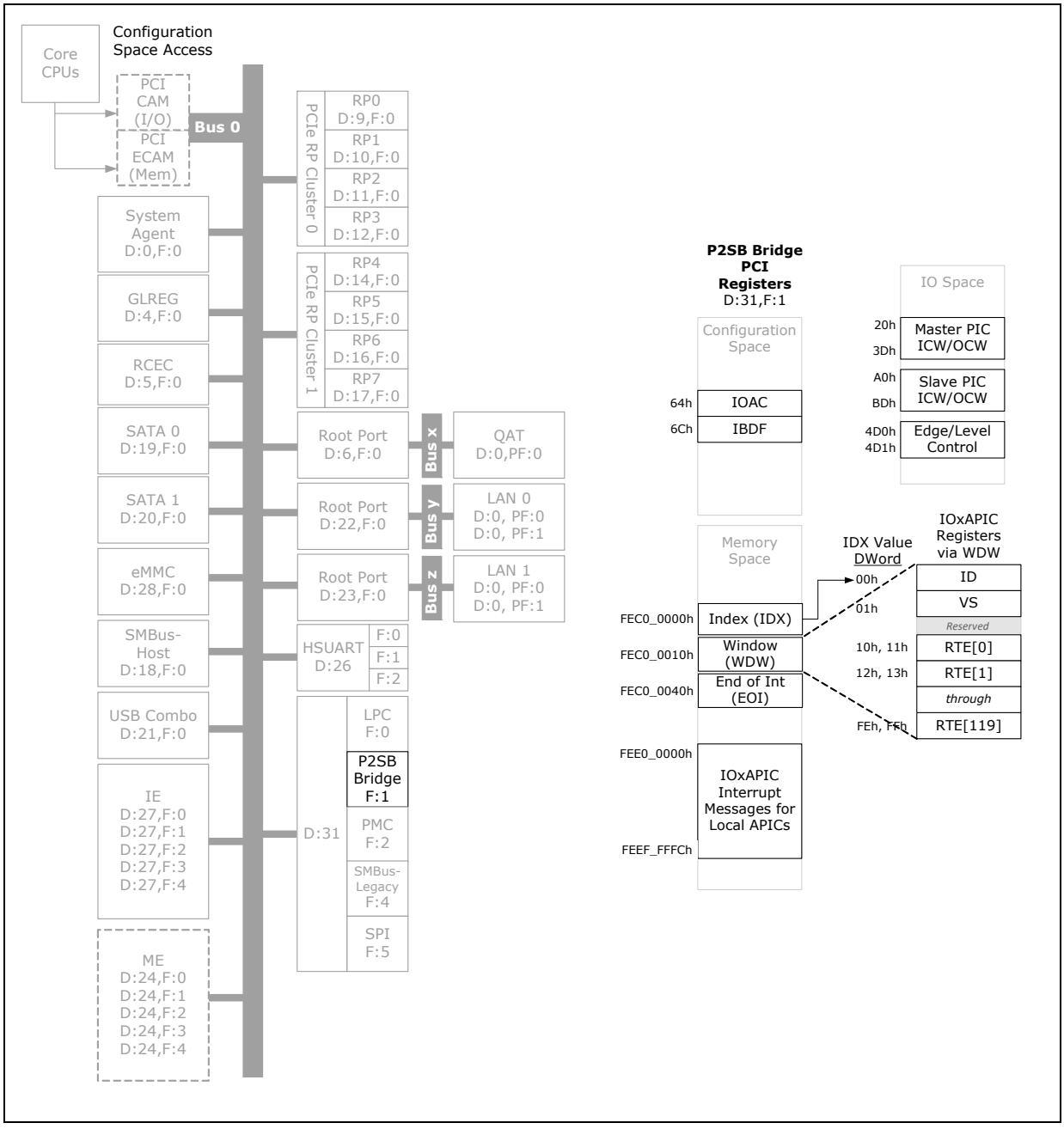
The 32-bit IOxAPIC EOI register is in Host Memory Space at address FEC0_0040h. The SoC P2SB Bridge PCI device (B:0, D:31, F:1) acts on the IOxAPIC's behalf. When a Host memory Write is issued to the 32-bit EOI register, the P2SB Bridge I/O checks the lower 8 bits written to this register and compares them with the Vector (VCT) field for each of the Redirection Table Entries (RTE 0 through RTE 119) in the Redirection Table.

When a match is found, the P2SB Bridge informs the IOxAPIC to clear the Remote IRR (RIRR) bit for that entry. If multiple entries have the same vector, each of those entries has its RIRR bit cleared.

## 21.4 Register Map

Figure 21-4 on page 650 shows the associated registers from a system software viewpoint.

**Figure 21-4. Register Map**



### 21.4.1 PCI Configuration and Capabilities

#### Table 21-5. Configuration and Capabilities Register Map

| Offset from Configuration Space of B0:D31:F1 (hex) | Register Short Name | Description |
|---|---|---|
| 64 | IOAC | IOxAPIC Configuration |
| 6C | IBDF | IOxAPIC Bus:Device:Function |

## 21.4.2 Memory-Mapped Registers

The registers listed in Table 21-7 are described in detail in Chapter 64, "PIC and I/O APIC Controllers".

The IOxAPIC Configuration (IOAC) register is located in Host Configuration Space Bus 0, Device 31 (decimal), Function 1, offset 0x64. When Address Enable (AE) is set in IOAC bit 8, IOAC bits 7:0 define bits 19:12 of the Host Memory Space address used to access the IOxAPIC memory-mapped registers.

Table 21-6 shows the resulting memory addresses when bit 8 of the IOAC register is zero, the default register setting. The default setting of IOAC enables compatibility with the default setting of prior Intel products.

**Table 21-6. IOxAPIC Register Access Window in Host Memory Space**

| Memory Address (hex) | Register Short Name | Register Long Name |
|---|---|---|
| FEC0_0000 | IDX | Index Register |
| FEC0_0010 | WDW | Window Register |
| FEC0_0040 | EOI | EOI Register |

Table 21-7 shows the IOxAPIC register accessed through the Window (WDW) Register for a given Index (IDX) register value. The register is indexed as 32-bit data entries as explained in Section 21.3.2, "Accessing the Redirection Table" on page 646.

**Table 21-7. IOxAPIC Registers**

| Index Register Value (hex) | Information Accessed in the Window Register Short Name | Information Accessed in the Window Register Long Name |
|---|---|---|
| 00 | ID | Identification Register |
| 01 | VS | Version |
| 02 through 0E | Reserved | Reserved |
| 10 through FE | RTE0 through RTE119 (decimal) | Redirection Table Entry 0 through 119 |

Table 21-8 shows the IOxAPIC Interrupt Message locations in Host Memory Space.

**Table 21-8. IOxAPIC Interrupt Messages for Local APICs in Host Memory Space**

| Memory Address (hex) | Register Short Name | Register Long Name |
|---|---|---|
| FEE0_0000<br>FEE0_0004<br>FEE0_0008<br>through FEEF_FFFC | - | IOxAPIC Interrupt Messages for processor Local APICs |

## 21.4.3 I/O-Mapped Registers

### Table 21-9. 8259 (PIC) Registers in Host I/O Space

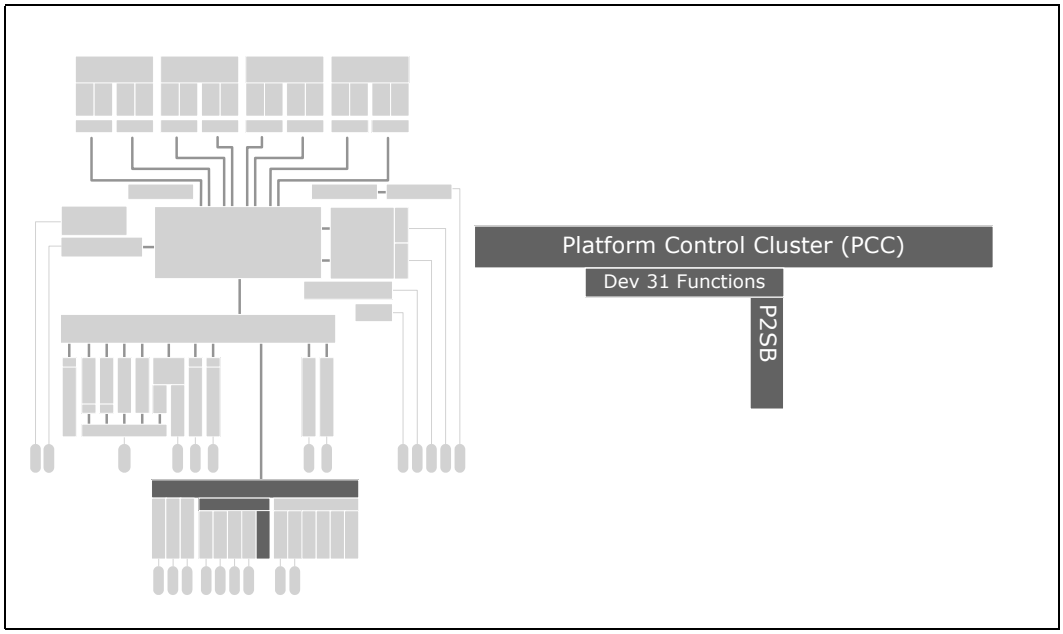
| I/O Port (hex) | Alias I/O Ports (hex) | Register Short Name | Register Name and Function |
|---|---|---|---|
| 20 | 24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch | MICW1 | Master Initialization Command Word 1 |
| | | MOCW2 | Master Operational Control Word 2 |
| | | MOCW3 | Master Operational Control Word 3 |
| 21 | 25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh | MICW2 | Master Initialization Command Word 2 |
| | | MICW3 | Master Initialization Command Word 3 |
| | | MICW4 | Master Initialization Command Word 4 |
| | | MOCW1 | Master Operational Control Word 1 |
| A0 | A4h, A8h, ACh, B0h, B4h, B8h, BCh | SICW1 | Slave Initialization Command Word 1 |
| | | SOCW2 | Slave Operational Control Word 2 |
| | | SOCW3 | Slave Operational Control Word 3 |
| A1 | A5h, A9h, ADh, B1h, B5h, B9h, BDh | SICW2 | Slave Initialization Command Word 2 |
| | | SICW3 | Slave Initialization Command Word 3 |
| | | SICW4 | Slave Initialization Command Word 4 |
| | | SOCW1 | Slave Operational Control Word 1 |
| 4D0 | - | ELCR1 | Master Edge/Level Control |
| 4D1 | - | ELCR2 | Slave Edge/Level Control |

§ §

# 22 Primary to Side Band (P2SB) Bridge

This chapter describes software access to the SoC Sideband registers.

**Figure 22-1. What is Covered in This Chapter**



## 22.1 Signal Descriptions

There are no signal pins or signal names associated with this chapter.

## 22.2 Feature List

- SoC Sideband Registers are accessed through Memory Space
- SBREG_BAR points to the 16-MB area in 64-bit-addressable memory

## 22.3 Architectural Overview

The Sideband Register access mechanism is configured at Bus 0, Device 31 (decimal), Function 1 of Configuration Space. Its PCI class designation is that of an unspecified other memory controller. SBREG_BAR and SBREG_BARH form the 64-bit memory Base Address Register (BAR) which points to a non-prefetchable, 16-MB region of memory-mapped space.

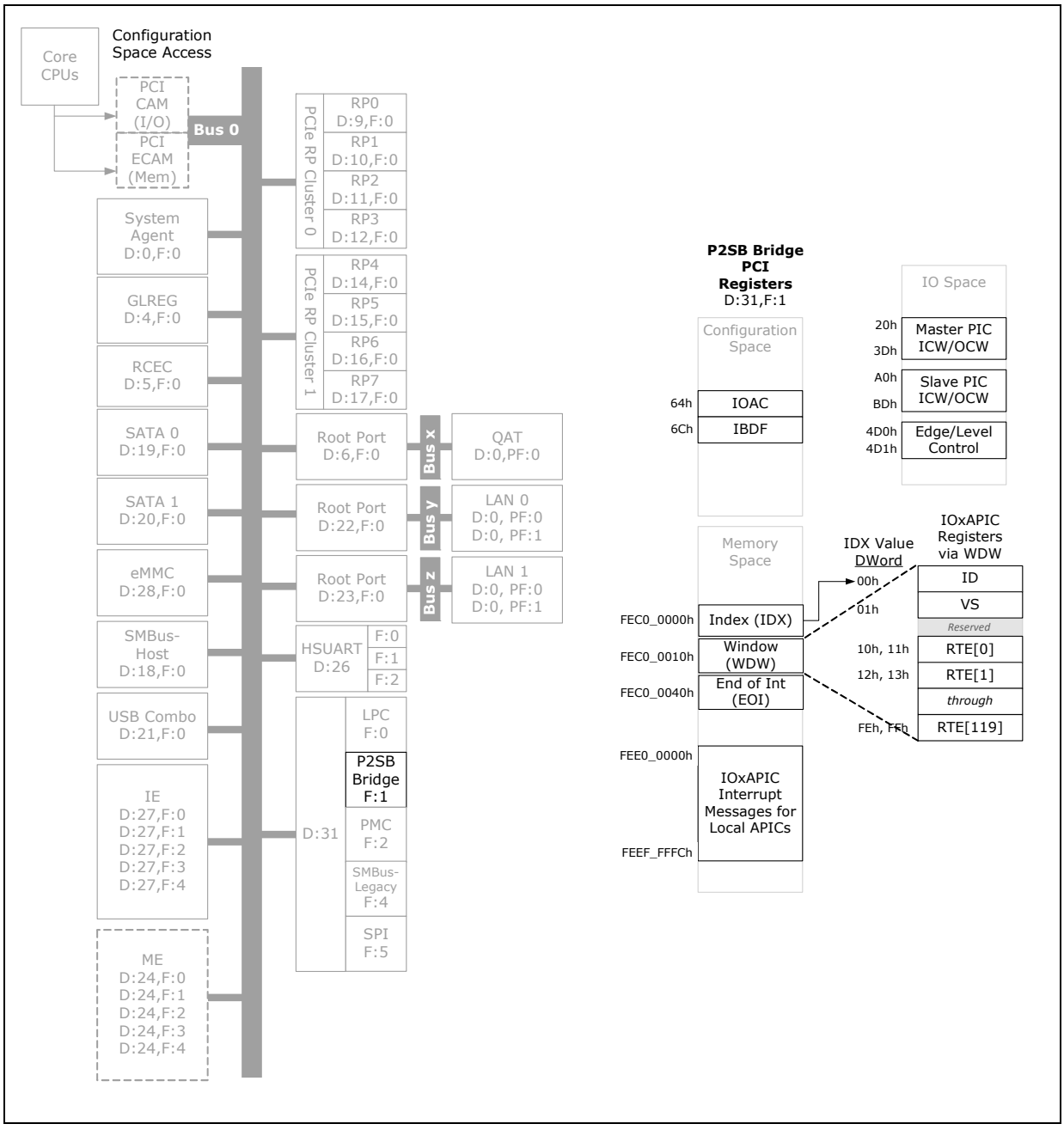
***Warning:*** All accesses to the SBREG_BAR region are DWORD aligned DWORDS. Writes larger than one DWORD are completed normally, but the write data beyond the first DWORD is discarded. Reads larger than one DWORD result in a Unsupported Request (UR) response. Byte enables within the DWORD are honored normally, enabling accesses of less than one DWORD.

The Sideband Register map is not provided in this chapter. Other chapters that refer to Sideband Registers provide the appropriate memory-mapped locations to access.

## 22.4 Register Map

Figure 22-2 shows the associated registers from a system software viewpoint.

**Figure 22-2. Register Map**



## 22.4.1 PCI Configuration and Capabilities

### Table 22-1. Configuration and Capabilities Register Map

| Offset from Configuration Space of B0:D31:F1 (hex) | Register Short Name | Description |
|---|---|---|
| **Standard Type 0 PCI Configuration Registers** | | |
| 0 | PCIID | PCI Identifier |
| 4 | PCICMD | PCI Command |
| 6 | PCISTS | PCI Status |
| 8 | PCIRID | Revision ID |
| 9 | PCICC | Class Code |
| D | PCIMLT | PCI Master Latency Timer |
| E | PCIHTYPE | PCI Header Type |
| 10 | SBREG_BAR | Sideband Register Access BAR |
| 14 | SBREG_BARH | Sideband Register BAR High DWORD |
| 2C | PCIHSS | PCI Subsystem Identifiers |
| **Device-Specific PCI Configuration Registers** | | |
| 50 | VBDF | VLW Bus:Device:Function |
| 52 | EBDF | ERROR Bus:Device:Function |
| 54 | RCFG | Routing Configuration |
| 60 | HPTC | High Performance Event Timer Configuration |
| 64 | IOAC | IOxAPIC Configuration |
| 6C | IBDF | IOxAPIC Bus:Device:Function |
| E0 | P2SBC | P2SB Control |
| F0 | URES | Unsupported Request Error Status |
| F4 | UREC | Unsupported Request Error Control |
| F8 | MANID | Manufacturer’s ID |

### 22.4.2 Memory-Mapped Registers

The registers listed in Table 22-2 that rely on customer access to the SoC are described in various chapters of this document.

**Table 22-2. Relocatable Memory-Mapped Registers**

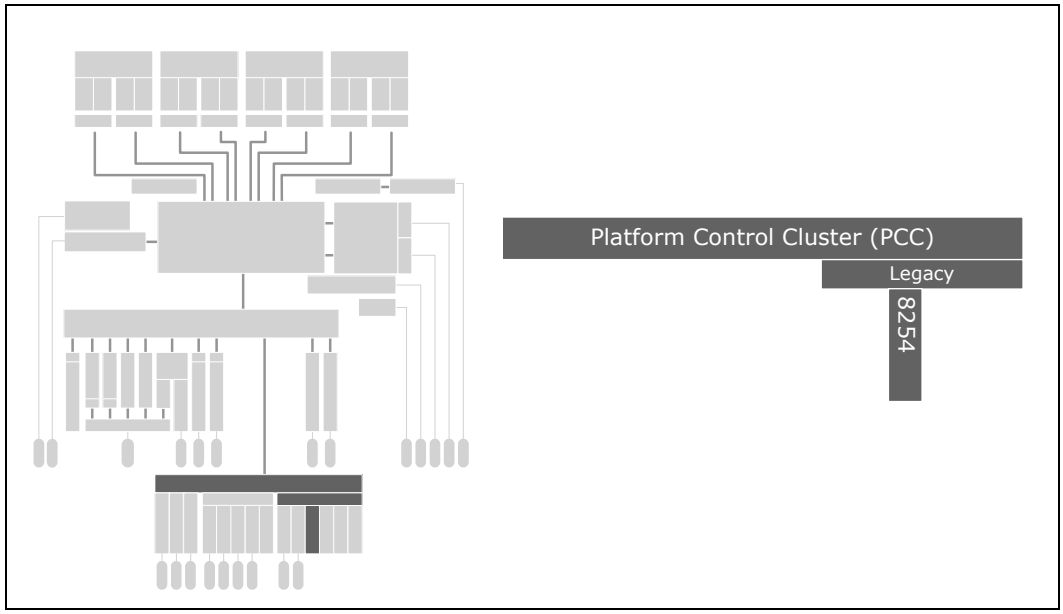
| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register |
|---|---|---|
| SBREG_BAR<br>SBREG_BARH | 0000 | First accessible Sideband Register |
| | 0004 | Second accessible Sideband Register |
| | 0008 | Third accessible Sideband Register |
| | through | |
| | FFFC | Last accessible Sideband Register |

§ §

# 23 Programmable Interval Timer (PIT)

This chapter describes the Programmable Interval Timer (PIT). The design is based on the legacy 8254 PIT device.

**Figure 23-1. What is Covered in This Chapter**



## 23.1 Signal Descriptions

There are no signal pins or signal names associated with this chapter.

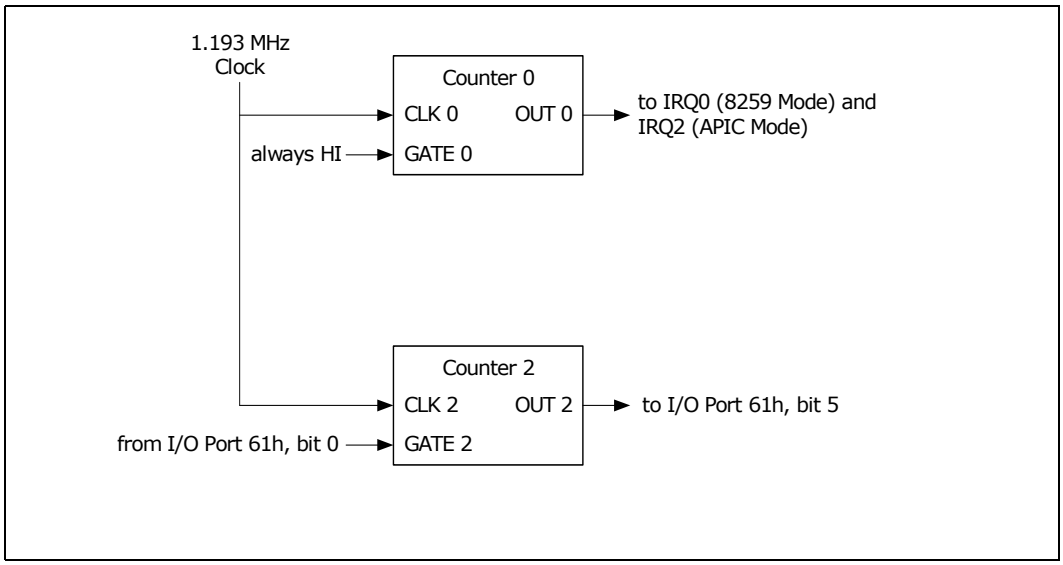
## 23.2 Feature List

- Legacy 8254 PIT functions.

## 23.3 Architectural Overview

The SoC contains two counters that have fixed uses. All registers and functions associated with the 8254 timers are in the Core power well. The 8254 unit is clocked by a 1.193-MHz clock derived from 25-MHz internal ISCLK clock. Figure 23-2 shows how the 8254 PIT is implemented in the SoC.

**Figure 23-2. Programmable Interval Timer (PIT) Implementation**



### 23.3.1 Counter 0, System Timer

Counter 0 functions as the system timer by controlling the state of IRQ0 and is typically programmed for Mode 3 operation. See Table 23-1, “Counter Operating Modes” on page 662. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value 1 counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle, alternately asserting and negating IRQ0.

### 23.3.2 Counter 2

Counter 2 was used in the original IBM PC to drive a speaker pin. The SoC does not have a dedicated speaker pin. The counter must be enabled by a write to TIM_CNT2_EN, bit 0 of the NMI_STS_CNT register at I/O Port 061h.

## 23.4 Counter Mode Definitions

Table 23-1 lists the six operating modes for the interval counters.

**Table 23-1. Counter Operating Modes**

| Mode | Function | Description |
|---|---|---|
| 0 | Out signal on end of count (=0) | Output is 0. When count goes to 0, output goes to 1 and stays at 1 until counter is reprogrammed. |
| 1 | Hardware retriggerable one-shot | Output is 0. When count goes to 0, output goes to 1 for one clock time. |
| 2 | Rate generator (divide by n counter) | Output is 1. Goes to 0 for 1 clock time, then back to 1 and counter is reloaded. |
| 3 | Square wave output | Output is 1. Output goes to 0 when counter rolls over, and counter is reloaded. Output goes to 1 when counter rolls over, and counter is reloaded, and so on |
| 4 | Software triggered strobe | Output is 1. Output goes to 0 when count expires for one clock time. |
| 5 | Hardware triggered strobe | Output is 1. Output goes to 0 when count expires for one clock time. |

## 23.5 Timer Programming

### 23.5.1 Writing to the Interval Timer

The counter/timers are programmed in the following fashion:

1. Write a control word to select a counter.
2. Write an initial count for that counter.
3. Load least and/or most significant bytes (as required by Control Word Bits 5, 4) of the 16-bit counter.
4. Repeat with other counters.

Only two conventions need to be observed when programming the counters.

1. For each counter, control word must be written before the initial count is written.
2. Initial count must follow the count format specified in the control word (least significant byte only, most significant byte only, or least significant byte, and then most significant byte).

A new initial count may be written to a counter at any time without affecting the counter's programmed mode. Counting is affected as described in the mode definitions. The new count must follow the programmed count format.

If a counter is programmed to read/write two-byte counts, the following precaution applies – a program must not transfer control between writing the first and second byte to another routine which also writes into that same counter. Otherwise, the counter will be loaded with an incorrect count.

The Timer Control Word (TCW) Register at I/O Port 43h controls the operation of all three counters. Several commands are available:

- Control Word Command. Specifies which counter to read or write, the operating mode, and the count format (binary or BCD). See Section 23.5.2.1, “Simple Read” on page 663.
- Counter Latch Command. Latches the current count so that it can be read by the system. The countdown process continues. See Section 23.5.2.2, “Counter Latch Command” on page 663.
- Read Back Command. Reads the count value, programmed mode, the current state of the OUT pins, and the state of the Null Count Flag of the selected counter. See Section 23.5.2.3, “Read Back Command” on page 664.

## 23.5.2 Reading from the Interval Timer

It is often desirable to read the value of a counter without disturbing the count in progress. There are three methods for reading the counters—a simple read operation, counter Latch command, and the Read-Back command. Each is explained below.

With the simple read and counter latch command methods, the count must be read according to the programmed format; specifically, if the counter is programmed for two byte counts, two bytes must be read. The two bytes do not have to be read one right after the other. Read, write, or programming operations for other counters may be inserted between them.

### 23.5.2.1 Simple Read

The first method is to perform a simple read operation. The counter is selected through C0_CAPR at I/O Port 40h (Counter 0) or C2_CAPR at I/O Port 42h (Counter 2).

*Note:* Performing a direct read from the counter does not return a determinate value, because the counting process is asynchronous to read operations. However, in the case of Counter 2, the count can be stopped by writing to the TIM_CNT2_EN bit, bit 0 of the NMI_STS_CNT register at I/O Port 061h.

### 23.5.2.2 Counter Latch Command

The Counter Latch command (CLC), written to I/O Port 43h, latches the count of a specific counter at the time the command is received. This command is used to ensure that the count read from the counter is accurate, particularly when reading a two-byte count. The count value is then read from each counter's Count register as was programmed by the Control register.

The count is held in the latch until it is read or the counter is reprogrammed. The count is then unlatched. This allows reading the contents of the counters on the fly without affecting counting in progress. Multiple Counter Latch Commands may be used to latch more than one counter. Counter Latch commands do not affect the programmed mode of the counter in any way.

If a Counter is latched and then, some time later, latched again before the count is read, the second Counter Latch command is ignored. The count read is the count at the time the first Counter Latch command was issued.

### 23.5.2.3 Read Back Command

The Read Back command (RBC), written to I/O Port 43h, latches the count value, programmed mode, and current states of the OUT pin and Null Count flag of the selected counter or counters. The value of the counter and its status may then be read by I/O access to the counter address.

The Read Back command may be used to latch multiple counter outputs at one time. This single command is functionally equivalent to several counter latch commands, one for each counter latched. Each counter's latched count is held until it is read or reprogrammed. Once read, a counter is unlatched. The other counters remain latched until they are read. If multiple count Read Back commands are issued to the same counter without reading the count, all but the first are ignored.

The Read Back command may additionally be used to latch status information of selected counters. The status of a counter is accessed by a read from that counter's I/O Port address. If multiple counter status latch operations are performed without reading the status, all but the first are ignored.

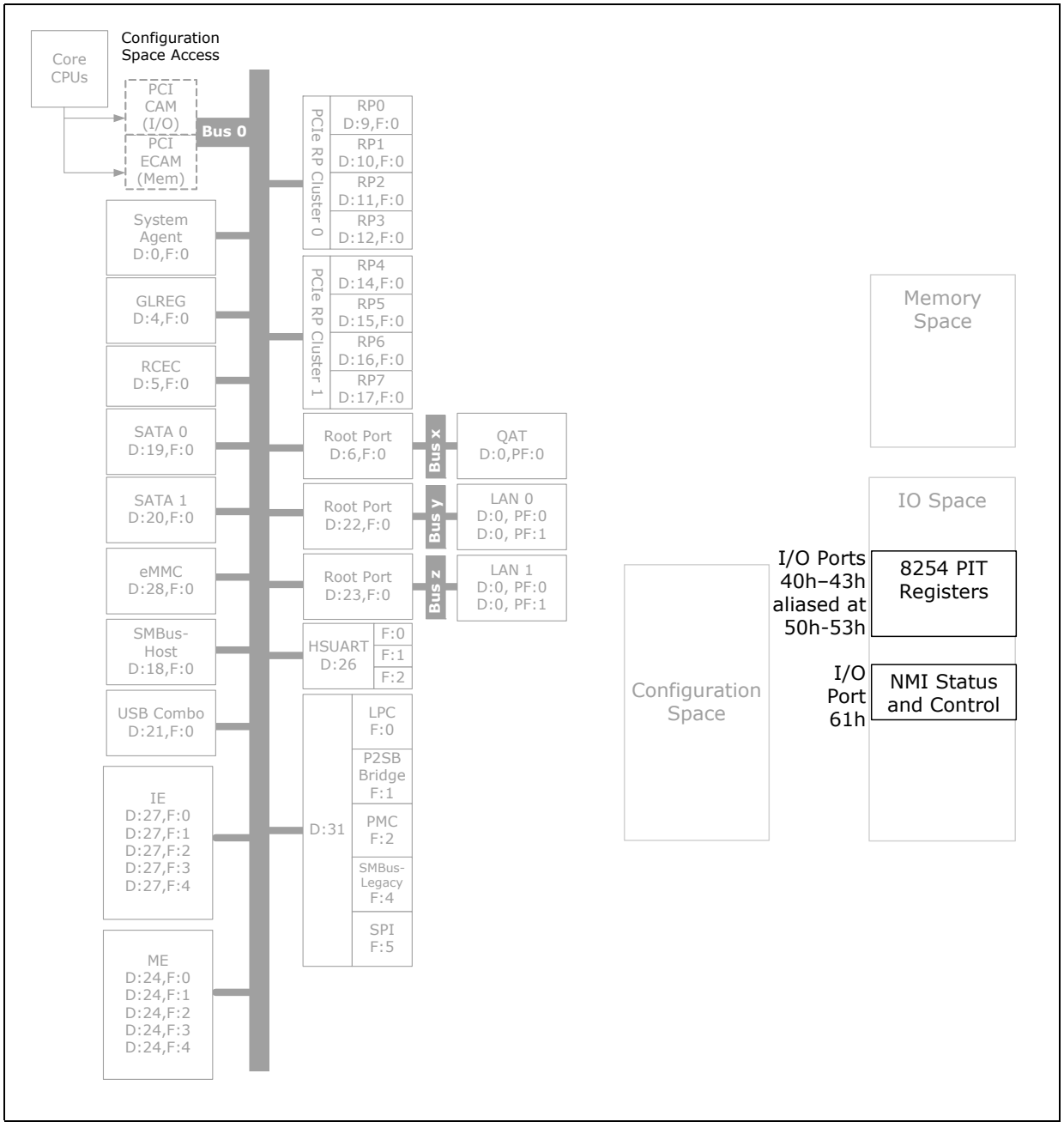
Both count and status of the selected counters may be latched simultaneously. This is functionally the same as issuing two consecutive, separate Read Back commands. If multiple count and/or status Read Back commands are issued to the same counters without any intervening reads, all but the first are ignored.

If both count and status of a counter are latched, the first read operation from that counter returns the latched status, regardless of which was latched first. The next one or two reads, depending on whether the counter is programmed for one or two type counts, returns the latched count. Subsequent reads return unlatched count.

## 23.6 Register Map

Figure 23-3 on page 665 shows the associated registers from a system software viewpoint.

**Figure 23-3. Register Map**



## 23.6.1 I/O-Mapped Registers

### Table 23-2. Fixed-Address I/O-Mapped Registers

| I/O Address and (Alias) | I/O Port Name | Access | Register Short Name | Register Long Name |
|---|---|---|---|---|
| 0x40 (0x50) | Port 40h | RO | C0_ITSBFR | Counter 0 - Interval Timer Status Byte Format Register |
| 0x41 (0x51) | *Reserved* | | *Reserved* | *Reserved* |
| 0x42 (0x52) | Port 42h | RO | C2_ITSBFR | Counter 2 - Interval Timer Status Byte Format Register |
| 0x43 (0x53) | Port 43h | WO | TCW | [1]Timer Control Word Register |
| 0x40 (0x50) | Port 40h | RW | C0_CAPR | Counter 0 - Counter Access Ports Register |
| 0x42 (0x52) | Port 42h | RW | C2_CAPR | Counter 2 - Counter Access Ports Register |
| 0x43 (0x53) | Port 43h | WO | RBC | [2]Read Back Command |
| 0x43 (0x53) | Port 43h | WO | CLC | [3]Counter Latch Command |
| 0x61 | Port 61 | RW | NMI_STS_CNT | NMI Status and Control |

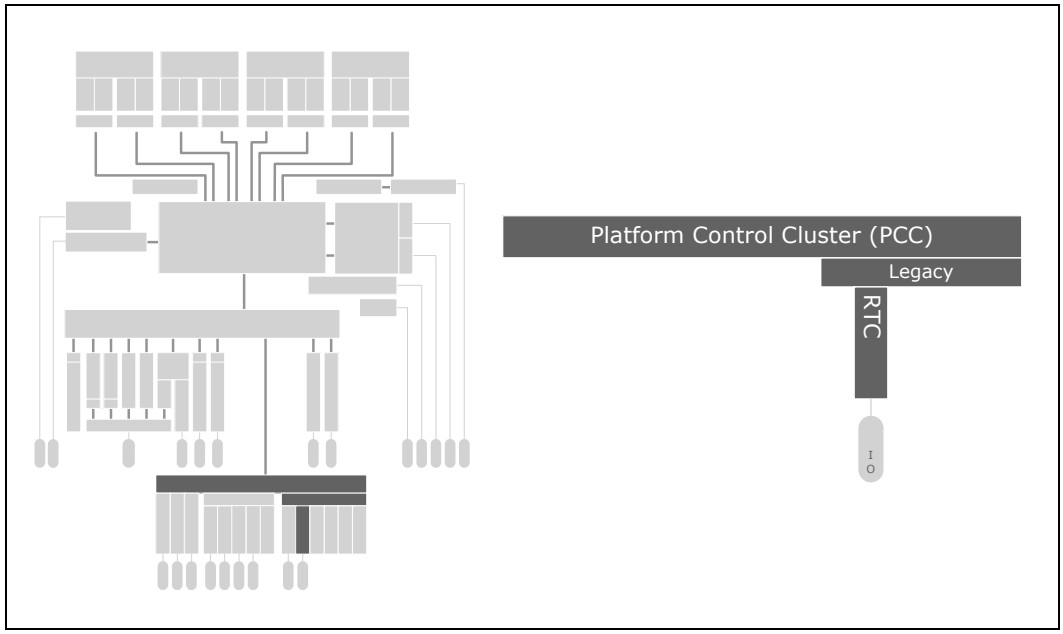
1. When data bits 7:6 are 00 or 10. The setting 01 is reserved. Also, data bits 5:4 must be 01, 10, or 11.
2. When data bits 7:6 are 11.
3. When data bits 7:6 are 00 or 10. The setting 01 is reserved. Also, data bits 5:4 must be 00.

§ §

# 24 Real Time Clock (RTC)

This chapter describes the SoC Real Time Clock (RTC).

**Figure 24-1. What is Covered in This Chapter**



## 24.1 Signal Descriptions

The signal descriptions are shown in Table 24-1. The signals associated with the RTC circuitry are shown as well as signal that reside in the SoC RTC power well. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing". The Direction/Type column of Table 24-1 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.

**Table 24-1. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| RTC_X1_PAD | I,O | No | **Crystal Input 1:** This signal is connected to the 32.768 KHz crystal (Max 50K ESR). If using an external oscillator, the RTCX1 Vih must be within the range of 0.8V to 1.5V (1.5V max). |
| RTC_X2_PAD | I,O | No | **Crystal Input 2:** This signal is connected to the 32.768 KHz crystal (Max 50K ESR). If using an external oscillator, RTCX2 should be left floating. |
| BVCCRTC_EXTPAD | I,O | No | **Internal VRM Filter:** Connect filter capacitor between this signal and ground. |
| COREPWROK | I | No | **Core Power OK:** When asserted, this active-high input signal indicates that all of the platform board Voltage Regulators (VRs) supplying the SoC core power rails have been stable for at least 10 ms.<br>COREPWROK can be driven asynchronously. When COREPWROK is de-asserted, the SoC asynchronously asserts the active-low PMU_PLTRST_N signal. |
| INTRUDER_N | I | No | **Intruder Detect:** This signal can be used to disable the system if the box is detected open. |
| RSMRST_N | I | No | **Resume (SUS Power Well) Reset**: When asserted by the platform board, this active-low input signal resets the SoC SUS Power Well.<br>An external RC circuit is required to guarantee that the SUS well power (VNN, 1.05V, 1.8V, and 3.3V) is valid for a minimum time prior to RSMRST_N signal going high (See Figure 33-1). |
| RTEST_N | I | No | **RTC Battery Test:** An external RC circuit creates a time delay for the signal such that it will go high (de-assert) sometime after the battery voltage is valid. If the battery is missing/weak, this signal appears low (asserted) at boot just after the suspend power rail (V3P3) is up since it will not have time to meet Vih when V3P3A is high. Upon booting, BIOS should recognize that RTCRCT# was asserted and clear CMOS RAM.<br>***Note:*** Unless CMOS is being cleared (only to be done in the G3 power state) or the battery is low, the signal input must always be high when all other RTC power planes are on. |
| SRTCRST_N | I | No | **RTC Reset:** An external RC circuit creates a time delay for the signal such that it will go high (de-assert) sometime after the battery voltage is valid. When asserted, this signal resets all register bits in the RTC well.<br>***Notes:***<br>1. Unless registers are being cleared (only to be done in the G3 power state), the signal input must always be high when all other RTC power planes are on.<br>2. In the case where the RTC battery is dead or missing on the platform, the signal should go high (de-asserted) before the RSMRST_N signal goes high (de-asserted). |

## 24.2 Feature List

- Maintaining power on VCCRTC_3P3_G3 (pin AU48) to the RTC when in the G3 state (e.g., coin-cell battery implementation) is optional.
- Legacy RTC features.

## 24.3 Architectural Overview

The Real Time Clock (RTC) module provides a battery backed-up date and time keeping device with two banks of static RAM with 128 bytes each, although the first bank has 114 bytes for general-purpose usage.

Three interrupt features are available:

- Time-of-day alarm with once-a-second to once-a-month range
- Periodic rates of 122 µs to 500 ms
- End-of-update cycle notification

Seconds, minutes, hours, days, day of week, month, and year are counted. Daylight savings compensation is optional. The hour is represented in twelve or twenty-four hour format and data can be represented in binary-coded decimal BCD or binary format.

The design is meant to be functionally compatible with the Motorola MS146818B. The time keeping comes from a 32.768 kHz oscillating source, which is divided to achieve an update every second. The RTC clock is not subject to any internal clock gating.

The lower 14 bytes of the lower static RAM block have very specific functions. The first ten are for time and date information. The next four (0Ah to 0Dh) are registers, which configure and report RTC functions.

The time and calendar data should match the data mode (BCD or binary) and hour mode (12 or 24 hour) as selected in register B. The programmer MUST make sure that data stored in these registers within the reasonable values ranges and represents a possible date and time. The exception to these ranges is to store a value of C0 - FF in the Alarm bytes to indicate a “don’t care” situation.

All Alarm conditions must match in order to trigger an Alarm Flag, which also triggers an Alarm Interrupt if enabled. The SET bit in register B should be 1 while programming these locations to avoid clashes with update cycles.

Access to time and date information is done through the static RAM locations. If a static-RAM read from the ten time and date bytes is attempted during an update cycle, the value read will not necessarily represent the true contents of those locations. Any static-RAM writes under the same conditions will be ignored.

*Note:* The leap year determination that adds a 29th day to the month of February does not take into account the end-of-the-century exceptions. The logic simply assumes that all years divisible by four are leap years. According to the Royal Observatory Greenwich, years that are divisible by 100 are typically not leap years. In every fourth century (years divisible by 400, like 2000), the 100-year-exception is over-ridden and a leap-year occurs. Note that the year 2100 will be the first time in which the current RTC implementation would incorrectly calculate the leap-year.

*Note:* The term RSMRST_N used within this chapter includes both the CF9h (in I/O Space) software-generated hardware reset and the RSMRST_N signal pin.

## 24.4 I/O Registers

The RTC internal registers and its static RAM are organized as two banks of 128 bytes each, called the Standard Bank and the Extended Bank:

- The first 14 bytes of the Standard Bank contain the RTC time and date information along with four registers: A; B; C; D. These four registers are used for configuration of the RTC but not physically exist in the static RAM.
- The Extended Bank, also called the "upper 128-byte bank," contains a full 128 bytes of battery-backed static RAM, and is accessible even when the RTC module is disabled via the RTC configuration register.

**Table 24-2. RTC Standard and Extended Static RAM Banks**

| Index (hex) | Register Short Name | Register Long Name |
|---|---|---|
| **Standard Bank** | | |
| 00 | Sec | Seconds |
| 01 | Sec_Alarm | Seconds Alarm |
| 02 | Minutes | Minutes |
| 03 | Minutes_Alarm | Minutes Alarm |
| 04 | Hours | Hours |
| 05 | Hours_Alarm | Hours Alarm |
| 06 | Day_of_Week | Day of Week |
| 07 | Day_of_Month | Day of Month |
| 08 | Month | Month |
| 09 | Year | Year |
| 0A | Register_A | Register A |
| 0B | Register_B | Register B - General Configuration |
| 0C | Register_C | Register C - Flag Register |
| 0D | Register_D | Register D - Flag Register |
| 0E - 7F | 114 Bytes of User RAM | |
| **Extended Bank** | | |
| 00 - 7F | 128 Bytes of User RAM | |

The two banks are accessible using an index mechanism. All data movement between the host CPU and the Real-Time Clock is done through registers mapped to Host Root I/O Space. The I/O register map appears below in Table 24-4, "Fixed-Address I/O-Mapped Registers".

The simplest way of reading the RTC registers is to:

- Use register pair located at I/O locations 74h and 75h to access the Standard Bank.
- Use register pair located at I/O locations 72h and 73h to access the Extended Bank.

### 24.4.1 Alternate-Access Mode

The NMI Enable bit located at I/O Port location 70h, bit 7, is best accessed ONLY when the chipset is put into Alternate-Access Mode. This guarantees successful read and write cycles regardless of the settings of the other registers. Alternate-Access Mode is enabled by setting Alternate Access Mode Enable (AME), bit 17 of the 32-bit General Interrupt Control (GIC) register in relocatable Memory Space at offset D0_31FCh from SBREG_BAR. When AME is set, read-only registers can be written, and write-only registers can be read.

## 24.5 Update Cycles

An Update Cycle occurs once a second if the SET bit of register B is not asserted and the divide chain is properly configured. During this procedure:

- the stored time and date will be incremented
- overflow will be checked
- a matching alarm condition will be checked
- the time and date will be rewritten to the static RAM locations.

The Update Cycle starts at least 488 µs after the UIP bit of register A is asserted. The entire cycle takes no more than 1984 µs to complete. The time and date static RAM locations (0-9) are disconnected from the external access bus during this time.

To avoid update and data corruption conditions, external static RAM access to these locations can safely occur upon the detection of either of two conditions:

- When an updated-ended interrupt is detected, almost 999 ms is available to read and write the valid time and date data.
- If the UIP bit of Register A is detected to be low, there is at least 488 µs before the update cycle begins.

***Warning:*** The overflow conditions for leap years and daylight savings adjustments are based on more than one date or time item. To ensure proper operation when adjusting the time, the new time and data values should be set at least two seconds before one of these conditions (leap year, daylight savings time adjustments) occurs.

## 24.6 Interrupts

The real-time clock interrupt is internally routed to the I/O APIC or the 8259. It is mapped to interrupt vector 8. This interrupt is not shared with any other interrupt. IRQ8# from the Serialized Interrupt (SERIRQ) stream is ignored. However, the High Performance Event Timers (HPET) can also be mapped to IRQ8#. In this latter case, the RTC interrupt is blocked.

## 24.7 Lockable RAM Ranges

The RTCs battery-backed static RAM supports an 8-byte range (index 38h through 3Fh) for the Standard Bank and also one for the Extended Bank which can be locked via the RTC Configuration (RC) register. If the locking bit is set, the corresponding range in the static RAM bank will not be readable or writable. A write cycle to those locations has no effect. A read cycle to those locations will not return the location's actual value (the value is undefined).

Once a range is locked, the range can be unlocked only by a hard reset, which will invoke the UEFI IA Firmware and allow it to re-lock the static RAM range.

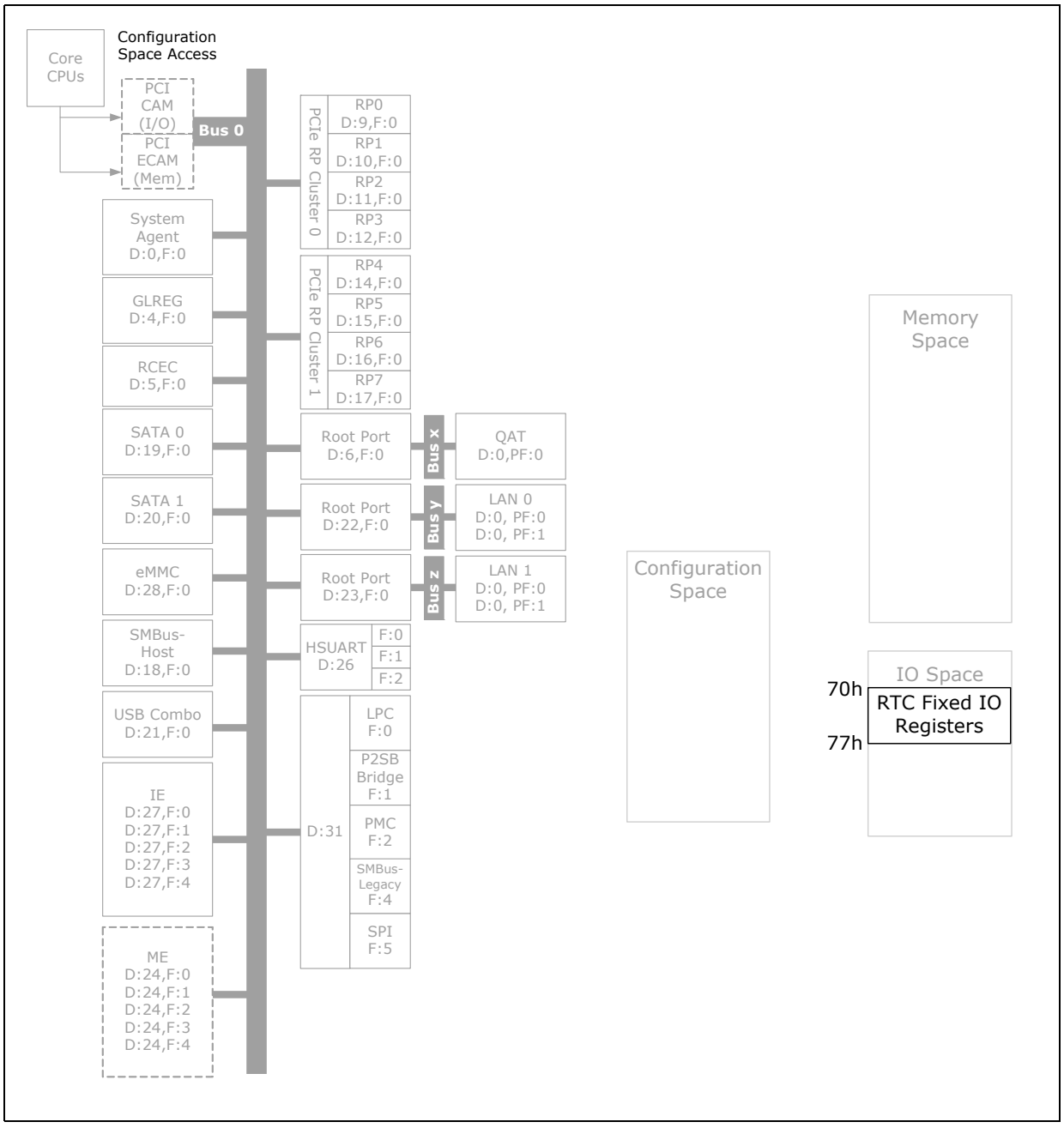
## 24.8 Month and Year Alarms

These functions are not implemented in the SoC.

## 24.9 Register Map

Table 24-2 shows how the RTC registers map to Host Root Space.

**Figure 24-2. Register Map**



### 24.9.1 PCI Configuration and Capabilities

There are no RTC registers in Host Root Configuration Space.

### 24.9.2 Memory-Mapped Registers

**Table 24-3. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| SBREG_BAR | D0_31FC | GIC | General Interrupt Control |
| SBREG_BAR | D1_3400 | RC | RTC Configuration |
| SBREG_BAR | D1_3414 | BUC | Backed Up Control |
| SBREG_BAR | D1_3F04 | UIPSMI | RTC Update In Progress SMI Control |
| SBREG_BAR | D1_3418 | RTCDCG | RTC Dynamic Clock Gating Control |

### 24.9.3 I/O-Mapped Registers

**Table 24-4. Fixed-Address I/O-Mapped Registers**

| I/O Locations (hex) | When Upper 128 Byte Enable (UE)[1] = 0 | Function |
|---|---|---|
| 70h and 74h | Also alias to 72h and 76h | Real-Time Clock (Standard RAM) Index Register<br>***Note:*** Writes to 72h, 74h, and 76h do not affect the NMI Enable bit (bit 7 of 70h) |
| 71h and 75h | Also alias to 73h and 77h | Real-Time Clock (Standard RAM) Target Register |
| 72h and 76h | | Extended RAM Index Register (if enabled) |
| 73h and 77h | | Extended RAM Target Register (if enabled) |

1. The Upper 128 Byte Enable (UE) register is bit 2 of the RTC Configuration (RC) register located in Host Root Memory Space at SBREG_BAR, offset D1_3400h.

§ §

# 25 High Precision Event Timer (HPET)

This chapter describes the High Precision Event Timer (HPET). One HPET Timer Block is provided which consists of a single counter that feeds eight comparators called Timer 0 through Timer 7. It is based on the *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a*.

**Figure 25-1. What is Covered in This Chapter**

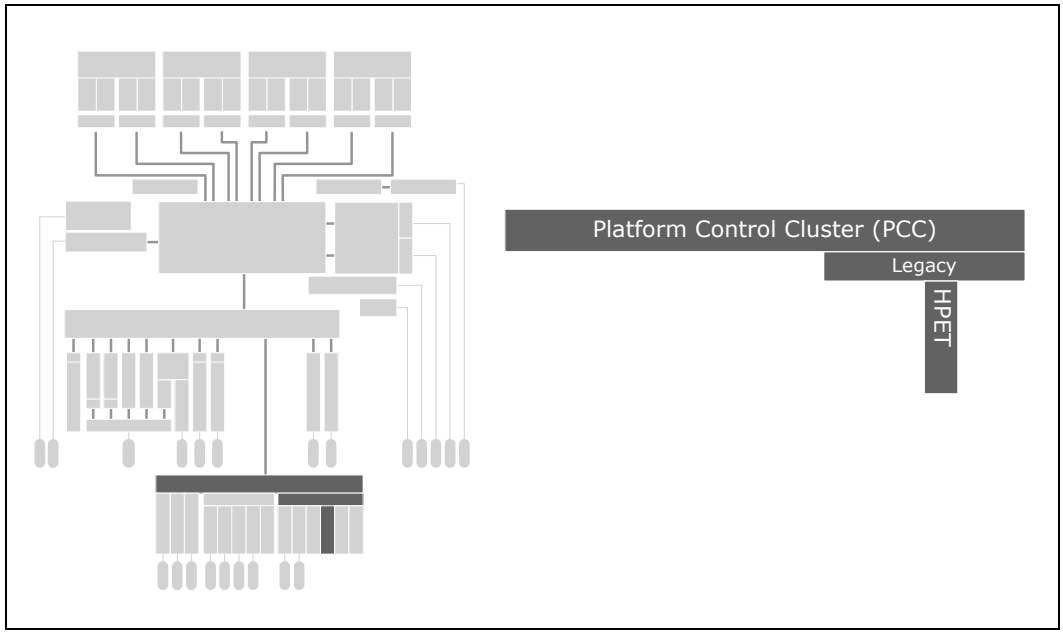



**Table 25-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| Intel HPET Specification | 1.0a | October 2004 | *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a* |

## 25.1 Signal Descriptions

There are no signal pins or signal names associated with this chapter.

## 25.2 Feature List

- The Time to Next Timer Event (TNTE) is conveyed internally from the HPET to the SoC Power Management Controller (PMC).
- HPET uses all 64-bit data as it allows two DW (QW) Register Write Access and Read Access.
- The HPET circuitry uses an internally-generated 24-MHz clock.

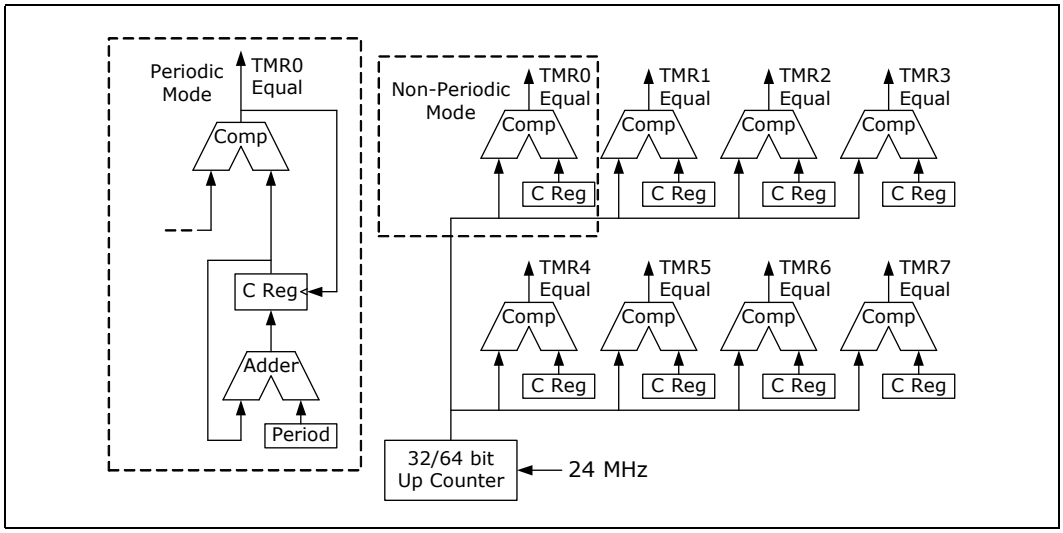
## 25.3 Architectural Overview

This function provides a set of timers that can be used by the operating system. The timers are defined such that the operating system may assign specific timers to be used directly by specific applications. Each timer can be configured to cause a separate interrupt.

The SoC provides eight timers. The timers are implemented as a single counter, and each timer has its own comparator and value register. The counter increases monotonically. Each individual timer can generate an interrupt when the value in its value register matches the value in the main counter. See Figure 25-2.

Timer 0 supports Periodic Interrupts.

**Figure 25-2. HPET Counter and Timers**




The registers associated with these timers are mapped to a range of fixed addresses in Host Root memory space. The UEFI IA Firmware (also known as BIOS) reports to the operating system the location of the register space via ACPI. The hardware can support an assignable decode space; however, UEFI IA Firmware sets this space to start at memory address 0xFED0_0000 prior to handing it over to the operating system. It is not expected that the operating system will move the location of these timers once it is set by UEFI IA Firmware.

### 25.3.1 Timer Accuracy

The timers are accurate over any 1-ms period to within 0.05% of the time specified in the timer resolution fields.

Within any 100-microsecond period, the timer reports a time that is up to two ticks too early or too late. Each tick is less than or equal to 100 ns; this represents an error of less than 0.2%.

The timer is monotonic. It does not return the same value on two consecutive reads unless the counter has rolled over and reached the same value.

The HPET main counter uses an internally-generated 24-MHz clock that is generated from the SoC 25-MHz crystal (ISCLK). The accuracy of the main counter is as accurate as the 25-MHz crystal that is used in the system. The HPET main counter increments in femtoseconds ($10^{-15}$ seconds). The internal 24-MHz clock has a period of 41,666,667 femtoseconds.

*Note:* Since the SoC does not support the processor C11 power state, the 32-kHz RTC clock is not used by the HPET circuitry.

### 25.3.2 Timer Off-Load

The HPET Timer Off-Load feature is not available with the SoC.

### 25.3.3 Interrupt Mapping

The interrupts associated with the various timers have several interrupt-mapping options.

Software should mask interrupts prior to clearing the LEG_RT_CNF bit. When reprogramming the HPET interrupt routing scheme, the LEG_RT_CNF bit of the General Configuration (GEN_CFG) Register, a spurious interrupt may occur. This is because the other source of the interrupt (the SoC 8254 Timer) may be asserted.

#### 25.3.3.1 Mapping Option #1 (Legacy Replacement Option)

In this case, the Legacy Replacement Rout (LEG_RT_CNF) bit of the General Configuration (GEN_CFG) Register is set. This forces the mapping found in Table 25-2.

**Table 25-2. Legacy Replacement Routing**

| Timer | 8259 Mapping | APIC Mapping | Comment |
|---|---|---|---|
| 0 | IRQ0 | IRQ2 | In this case, the 8254 Timer will not cause any interrupts |
| 1 | IRQ8 | IRQ8 | In this case, the RTC will not cause any interrupts. |
| 2 and 3 | Per IRQ Routing Field. | Per IRQ Routing Field | |
| 4, 5, 6, 7 | not available | not available | |

***Note:*** The Legacy Option does not preclude delivery of IRQ0/IRQ8 using processor interrupts messages.

### 25.3.3.2 Mapping Option #2 (Standard Option)

In this case, the Legacy Replacement Rout (LEG_RT_CNF) bit of the of the General Configuration (GEN_CFG) Register is 0. Each timer has its own routing control. The interrupts can be routed to various interrupts of the 8259 PIC or IOxAPIC. A capabilities field indicates which interrupts are valid options for routing. If a timer is set for edge-triggered mode, the timers should not be shared with any legacy interrupts.

For the SoC, the only supported interrupt values are as follows:

- Timer 0 and 1: IRQ20, 21, 22, and 23 (I/O APIC only).
- Timer 2: IRQ11 (8259 or I/O APIC) and IRQ20, 21, 22, and 23 (I/O APIC only).
- Timer 3: IRQ12 (8259 or I/O APIC) and IRQ 20, 21, 22, and 23 (I/O APIC only).

*Note:* Interrupts from Timers 4, 5, 6, and 7 can only be delivered using processor message interrupts.

### 25.3.3.3 Mapping Option #3 (Processor Message Option)

Here the interrupts are mapped directly to processor messages, also called FSB Interrupt Messages, without going to the 8259 PIC or IOxAPIC. To use this mode, the interrupt must be configured to edge-triggered mode. The TIMERn_FSB_EN_CNF bit of the Timer n Configuration and Capabilities (TMR0_CNF_CAP for example) Register must be set to enable this mode.

Software configures where the 32-bit message is written within memory-mapped space. Each of the eight timers has a 64-bit "Timer n FSB Interrupt Rout Register" (TMR0_FSB_INT_ROUT for example) for the 32-bit message address and the 32-bit message data.

*Note:* The processor message interrupt delivery option has HIGHER priority and is mutually exclusive to the standard interrupt delivery option. Thus, if the TIMERn_FSB_EN_CNF bit is set, the interrupts will be delivered directly to the processor, rather than by means of the APIC or 8259 PIC.

The processor message interrupt delivery can be used even when the legacy mapping is used.

## 25.3.4 Periodic Versus Non-Periodic Modes

### 25.3.4.1 Non-Periodic (One-Shot) Mode

This mode change can be thought of as creating a one-shot timer.

Timer 0 is configurable to 32- (default) or 64-bit mode, whereas Timers 1 through timer 7 only support 32-bit mode. All of the timers support Non-Periodic Mode. For details of this mode, see Section 2.3.9.2.1 of the *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a*.

***Warning:*** Software must be careful when programming the comparator registers. If the value written to the register is not sufficiently far in the future, then the counter may pass the value before it reaches the register and the interrupt will be missed. The UEFI IA Firmware should pass a data structure to the operating system to indicate that the operating system should not attempt to program the periodic timer to a rate faster than 5 microseconds.

### 25.3.4.2 Periodic Mode

Timer 0 is the only timer that supports Periodic Mode. For details of this mode, see Section 2.3.9.2.2 of the *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a*.

If the software resets the main counter, the value in the comparator's value register needs to reset as well. This can be done by setting TIMER0_VAL_SET_CNF, bit 6 of the Timer 0 Configuration and Capabilities (TMR0_CNF_CAP) Register. Again, to avoid race conditions, this should be done with the main counter halted. The following usage model is expected:

1. Software clears the ENABLE_CNF, bit 0 of the General Configuration (GEN_CFG) Register, to prevent any interrupts.
2. Software Clears the main counter by writing a value of 00h to it.
3. Software sets the TIMER0_VAL_SET_CNF bit.
4. Software writes the new value in the TIMER0_COMPARATOR_VAL register.
5. Software sets the ENABLE_CNF bit to enable interrupts.

The Timer 0 Comparator Value register cannot be programmed reliably by a single 64-bit write in a 32-bit environment, except if only the periodic rate is being changed during run-time. If the actual Timer 0 Comparator Value needs to be reinitialized, then the following software solution will always work, regardless of the environment:

1. Set TIMER0_VAL_SET_CNF bit.
2. Set the lower 32 bits of the Timer0 Comparator Value register.
3. Set TIMER0_VAL_SET_CNF bit.
4. Set the upper 32 bits of the Timer0 Comparator Value register.

### 25.3.5 Enabling the Timers

The UEFI IA Firmware or operating system Plug and Play (PnP) code should route the interrupts. This includes the Legacy Replacement Rout bit, Interrupt Rout bit (for each timer), and interrupt type (to select the edge or level type for each timer).

The Device Driver code should do the following for an available timer:

1. Set the Overall Enable bit (ENABLE_CNF), bit 0 of the General Configuration (GEN_CFG) Register.

If Timer 0 is being set up, program the Timer 0 Type field, bit 3 of the Timer 0 Configuration and Capabilities (TMR0_CNF_CAP) Register, to either Periodic Mode or Non-Periodic Mode.

2. Set the interrupt enable.
3. Set the comparator value.

### 25.3.6 Interrupt Levels

Interrupts directed to the internal 8259 PIC are active high. See Chapter 21, "PIC and I/O APIC Controllers" for information regarding the programming the polarity of 8259 PIC for detecting internal interrupts.

If the interrupts are mapped to the 8259 or I/O APIC and set for level-triggered mode, they can be shared with legacy interrupts. They may be shared although it is unlikely for the operating system to attempt to do this.

If more than one timer is configured to share the same IRQ using the TIMERn_INT_ROUT_CNF fields of the Timer n Configuration and Capabilities (TMR0_CNF_CAP for example) Register, then the software must configure the timers to level- triggered mode. Edge-triggered interrupts cannot be shared.

### 25.3.7 Handling Interrupts

Section 2.4.6 of the *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a* describes handling interrupts.

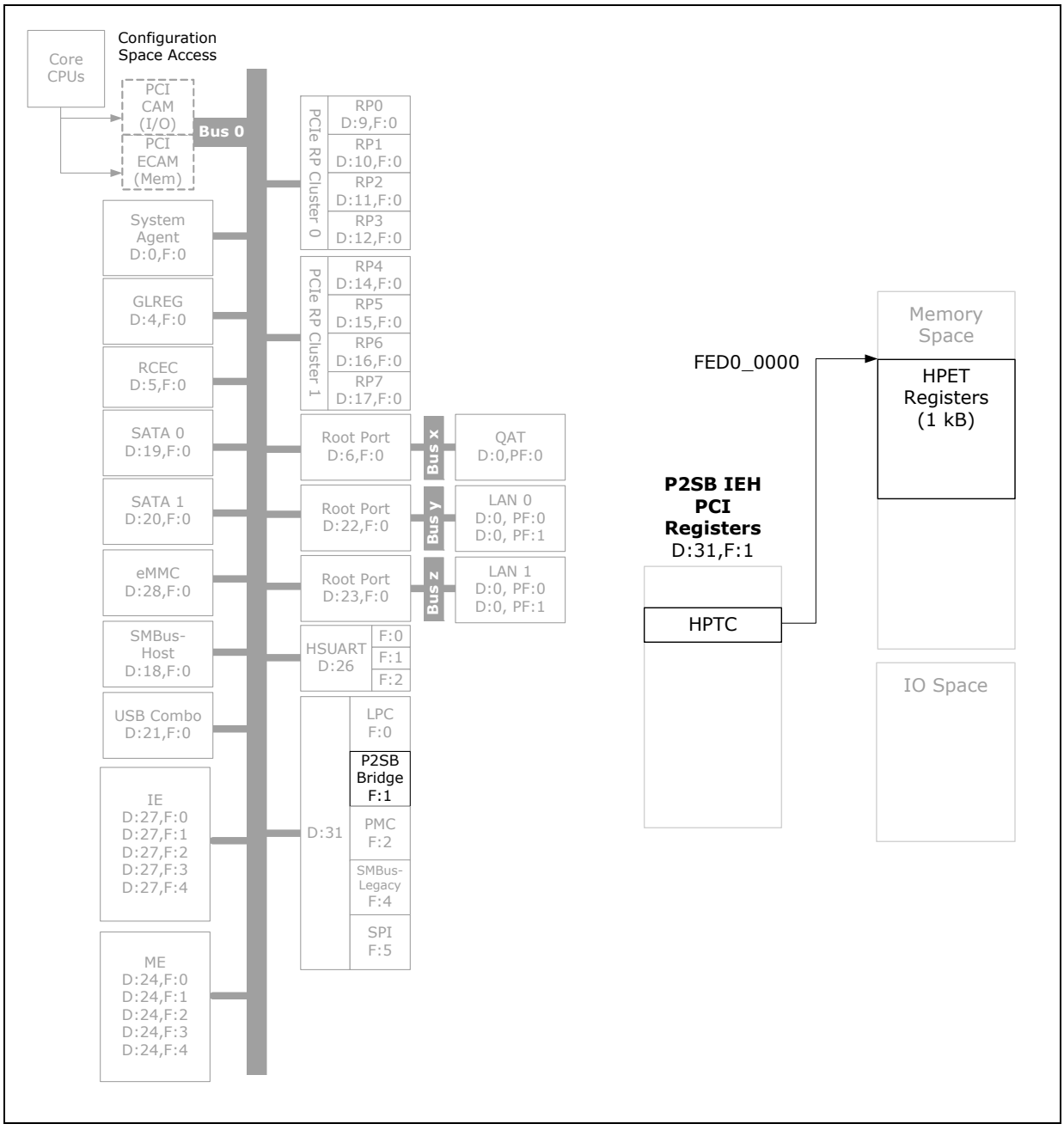
### 25.3.8 Issues Related to 64-Bit Timers with 32-Bit Processors

Section 2.4.7 of the *IA-PC HPET (High Precision Event Timers) Specification Revision 1.0a* describes issues related to 64-Bit timers with 32-Bit processors.

## 25.4 Register Map

Figure 25-3 shows the associated registers from a system software viewpoint.

**Figure 25-3. Register Map**



## 25.4.1 Memory-Mapped Registers

*Note:* The base memory address for the HPET registers is determined by the High Performance Event Timer Configuration (HPTC) register in configuration space at offset 60h of the P2SB Bridge (Bus 0, Device 31, Function 1). See Figure 25-3. Even though UEFI IA Firmware can configure the HPET timers to be located at one of four locations in memory space, only the location starting with 0xFED0_0000 is validated by Intel. The other three memory locations are not documented in this document.

**Table 25-3. Fixed-Address Memory-Mapped Registers (Sheet 1 of 2)**

| Fixed Address in Memory Space | 64-bit Register Short Name | 64-bit Register Long Name |
|---|---|---|
| 0xFED00000 | GEN_CAP_ID | General Capabilities and ID Register |
| 0xFED00008 | *Reserved* | *Reserved* |
| 0xFED00010 | GEN_CFG | General Configuration Register |
| 0xFED00018 | *Reserved* | *Reserved* |
| 0xFED00020 | GEN_INT_STS | General Interrupt Status Register |
| 0xFED00028 through 0xFED000E8 | *Reserved* | *Reserved* |
| 0xFED000F0 | MAIN_CNTR | Main Counter Value |
| 0xFED000F8 | *Reserved* | *Reserved* |
| 0xFED00100 | TMR0_CNF_CAP | Timer 0 Configuration and Capabilities |
| 0xFED00108 | TMR0_CMP_VAL | Timer 0 Comparator Value |
| 0xFED00110 | TMR0_FSB_INT_ROUT | Timer 0 FSB Interrupt Rout Register |
| 0xFED00118 | *Reserved* | *Reserved* |
| 0xFED00120 | TMR1_CNF_CAP | Timer 1 Configuration and Capabilities |
| 0xFED00128 | TMR1_CMP_VAL | Timer 1 Comparator Value |
| 0xFED00130 | TMR1_FSB_INT_ROUT | Timer 1 FSB Interrupt Rout Register |
| 0xFED00138 | *Reserved* | *Reserved* |
| 0xFED00140 | TMR2_CNF_CAP | Timer 2 Configuration and Capabilities |
| 0xFED00148 | TMR2_CMP_VAL | Timer 2 Comparator Value |
| 0xFED00150 | TMR2_FSB_INT_ROUT | Timer 2 FSB Interrupt Rout Register |
| 0xFED00158 | *Reserved* | *Reserved* |
| 0xFED00160 | TMR3_CNF_CAP | Timer 3 Configuration and Capabilities |
| 0xFED00168 | TMR3_CMP_VAL | Timer 3 Comparator Value |
| 0xFED00170 | TMR3_FSB_INT_ROUT | Timer 3 FSB Interrupt Rout Register |
| 0xFED00178 | *Reserved* | *Reserved* |
| 0xFED00180 | TMR4_CNF_CAP | Timer 4 Configuration and Capabilities |
| 0xFED00188 | TMR4_CMP_VAL | Timer 4 Comparator Value |
| 0xFED00190 | TMR4_FSB_INT_ROUT | Timer 4 FSB Interrupt Rout Register |
| 0xFED00198 | *Reserved* | *Reserved* |
| 0xFED001A0 | TMR5_CNF_CAP | Timer 5 Configuration and Capabilities |
| 0xFED001A8 | TMR5_CMP_VAL | Timer 5 Comparator Value |
| 0xFED001B0 | TMR5_FSB_INT_ROUT | Timer 5 FSB Interrupt Rout Register |
| 0xFED001B8 | *Reserved* | *Reserved* |

### Table 25-3. Fixed-Address Memory-Mapped Registers (Sheet 2 of 2)

| Fixed Address in Memory Space | 64-bit Register Short Name | 64-bit Register Long Name |
|---|---|---|
| 0xFED001C0 | TMR6_CNF_CAP | Timer 6 Configuration and Capabilities |
| 0xFED001C8 | TMR6_CMP_VAL | Timer 6 Comparator Value |
| 0xFED001D0 | TMR6_FSB_INT_ROUT | Timer 6 FSB Interrupt Rout Register |
| 0xFED001D8 | *Reserved* | *Reserved* |
| 0xFED001E0 | TMR7_CNF_CAP | Timer 7 Configuration and Capabilities |
| 0xFED001E8 | TMR7_CMP_VAL | Timer 7 Comparator Value |
| 0xFED001F0 | TMR7_FSB_INT_ROUT | Timer 7 FSB Interrupt Rout Register |
| 0xFED001F8 through 0xFED003F8 | *Reserved* | *Reserved* |

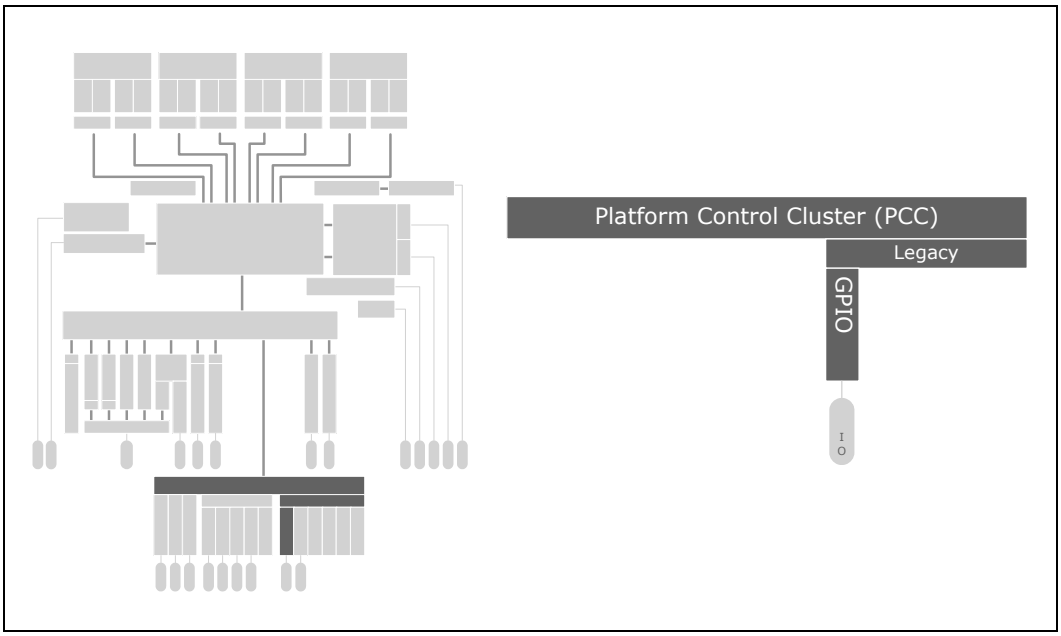
§ §

# 26 Customer General-Purpose I/O (GPIO)

A number of SoC signal pins have shared, native functions. These signals are called Configurable Flexible I/O (CFIO) signals. If the default setting is not needed for the customer board design, one of the other shared, native signals can be configured for the pin. The silicon pad connected to each CFIO signal pin contains a signal selector (also called MUX) that can be programmed. Of the SoC CFIO signal pins, the SoC provides 147 General-Purpose I/O (GPIO) signals that can be used by the SoC customer. These special CFIO signals are called Customer GPIOs. For example, the GPIO125 default function mode is eMMC clock. See details in Table 39-1, "Signal Pin Name Directory - Sorted by Name" on page 995.

- GPIO registers are accessible by the Host or IE through their MMIO space, using the P2SB (Primary to Sideband Bridge) (B:0 D:31 F:1) SBREG contents as the base address.
- NMI and SMI are capable only on selected GPIOs.
- SCI (GPE) and IOxAPIC interrupt are capable on all GPIOs.
- The Intel® ME assigns ownership of individual GPIO pads. The Intel® ME assigns ownership of individual GPIO pads to the HOST or IE or Intel® ME.
- Legacy I/O space control of GPIOs is not supported

**Figure 26-1. What is Covered in This Chapter**



## 26.1 GPIO Ball and Signal Names

The term "pad" is used in the register descriptions of the SoC. The pad is located on the SoC silicon. The signal is wired to a pin (a.k.a., ball) on the product package. The SoC pad control registers refer to the silicon pads, but in essence they apply to the signal at the pin. This chapter uses "pad" and "pin" interchangeably.

Table 26-1, "GPIO Ball and Pad Name" shows GPIO ball mapping to the signal names. The table is an extension to Table 39-1, "Signal Pin Name Directory - Sorted by Name" on page 995.

**Table 26-1. GPIO Ball and Pad Name (Sheet 1 of 5)**

| Number | Ball/Pin | Ball/Pin Name | Notes |
|---|---|---|---|
| 0 | BF70 | GPIO[0] | |
| 1 | BK69 | GPIO[1] | |
| 2 | BK71 | GPIO[2] | |
| 3 | BN44 | GPIO[3] | |
| 4 | BH50 | GPIO[4] | |
| 5 | BL48 | GPIO[5] | |
| 6 | BE52 | GPIO[6] | |
| 7 | BH46 | GPIO[7] | |
| 8 | BY55 | GPIO[8] | |
| 9 | CA54 | GPIO[9] | |
| 10 | CA63 | ESPI_CS_N[1] | |
| 11 | BY62 | ESPI_ALRT_N[1] | |
| 12 | BN56 | GPIO[12] | |
| 13 | BN66 | SMB_LAN_ALRT_N | |
| 14 | AV66 | LAN0_PORT0_SDP[0] | |
| 15 | AY66 | LAN0_PORT1_SDP[0] | |
| 16 | BA71 | LAN0_PORT0_SDP[1] | |
| 17 | BB72 | LAN0_PORT1_SDP[1] | |
| 18 | AC59 | LAN0_PORT0_SDP[2] | |
| 19 | AU65 | LAN0_PORT1_SDP[2] | |
| 20 | BA69 | LAN0_PORT0_SDP[3] | |
| 21 | W58 | LAN0_PORT1_SDP[3] | |
| 22 | AY70 | LAN1_PORT0_LED[0] | |
| 23 | AY72 | LAN1_PORT0_LED[1] | |
| 24 | AV59 | LAN0_PORT0_I2C_CLK | |
| 25 | AV56 | LAN0_PORT0_I2C_DATA | |
| 26 | BB66 | LAN0_PORT1_I2C_CLK | |
| 27 | BC69 | LAN0_PORT1_I2C_DATA | |
| 28 | AY59 | NCSI_RXD[0] | |
| 29 | AY63 | NCSI_CLK_IN | |
| 30 | BD72 | NCSI_RXD[1] | |
| 31 | BD65 | NCSI_CRS_DV | |

## Table 26-1. GPIO Ball and Pad Name (Sheet 2 of 5)

| Number | Ball/Pin | Ball/Pin Name | Notes |
|---|---|---|---|
| 32 | AU58 | NCSI_ARB_IN | |
| 33 | BD70 | NCSI_TX_EN | |
| 34 | BE66 | NCSI_TXD[0] | |
| 35 | AL56 | NCSI_TXD[1] | |
| 36 | AV63 | NCSI_ARB_OUT | |
| 37 | BE69 | LAN0_PORT0_LED[0] | |
| 38 | BF72 | LAN0_PORT0_LED[1] | |
| 39 | BA65 | LAN0_PORT1_LED[0] | |
| 40 | BG66 | LAN0_PORT1_LED[1] | |
| 41 | AP56 | PCIE_CLKREQ_N[0] | |
| 42 | BA58 | PCIE_CLKREQ_N[1] | |
| 43 | BH71 | PCIE_CLKREQ_N[2] | |
| 44 | BJ66 | PCIE_CLKREQ_N[3] | |
| 45 | BD61 | PCIE_CLKREQ_N[4] | |
| 46 | n/a | n/a | |
| 47 | BA54 | SVID_ALERT_N | SVID Pads |
| 48 | BE56 | SVID_DATA | |
| 49 | BE59 | SVID_CLK | |
| 50 | BA61 | THERMTRIP_N | TM Pads |
| 51 | BM71 | PROCHOT_N | |
| 52 | BL70 | MEMHOT_N | |
| 53 | AY56 | DFX_PORT_CLK[0] | DFX Set2 Pads |
| 54 | BB70 | DFX_PORT_CLK[1] | |
| 55 | BN68 | DFX_PORT[0] | |
| 56 | BN70 | DFX_PORT[1] | |
| 57 | BE63 | DFX_PORT[2] | |
| 58 | BP69 | DFX_PORT[3] | |
| 59 | BW68 | DFX_PORT[4] | |
| 60 | BH69 | DFX_PORT[5] | |
| 61 | BG63 | DFX_PORT[6] | |
| 62 | BT69 | DFX_PORT[7] | |
| 63 | BW67 | DFX_PORT[8] | |
| 64 | BD58 | ESPI_IO[0] | |
| 65 | BR67 | ESPI_IO[1] | |
| 66 | BV64 | ESPI_IO[2] | |
| 67 | BH65 | ESPI_IO[3] | |
| 68 | BJ72 | ESPI_CS_N[0] | |
| 69 | BY64 | ESPI_CLK | |
| 70 | BV66 | ESPI_RST_N | |
| 71 | BK61 | ESPI_ALRT_N[0] | |

## Table 26-1. GPIO Ball and Pad Name (Sheet 3 of 5)

| Number | Ball/Pin | Ball/Pin Name | Notes |
|---|---|---|---|
| 72 | BG65 | SPI_CS_N[0] | |
| 73 | BL66 | SPI_CS_N[1] | |
| 74 | BU63 | SPI_MOSI_IO[0] | |
| 75 | BV62 | SPI_MOSI_IO[1] | |
| 76 | BH61 | SPI_IO[2] | |
| 77 | BW61 | SPI_IO[3] | |
| 78 | CA61 | SPI_CLK | |
| 79 | BH58 | SUSPWRDNACK | |
| 80 | BG59 | PMU_SUSCLK | |
| 81 | BV60 | ADR_TRIGGER | |
| 82 | BW59 | PMU_SLP_S45_N | |
| 83 | BK65 | PMU_SLP_S3_N | |
| 84 | BR61 | PMU_WAKE_N | |
| 85 | CA59 | PMU_PWRBTN_N | |
| 86 | BL59 | PMU_RESETBUTTON_N | |
| 87 | BL63 | PMU_PLTRST_N | |
| 88 | BV58 | SUS_STAT_N | |
| 89 | BY58 | RSVD_BY58 | |
| 90 | BR65 | SATA0_LED_N | |
| 91 | BV55 | SATA1_LED_N | |
| 92 | CA57 | SATA_PDETECT[0] | |
| 93 | BN63 | SATA_PDETECT[1] | |
| 94 | BU57 | SATA0_SDOUT | |
| 95 | BW54 | SATA1_SDOUT | |
| 96 | BV53 | UART1_RXD | |
| 97 | BN59 | UART1_TXD | |
| 98 | BW52 | PCIE_CLKREQ_N[5] | |
| 99 | BV51 | PCIE_CLKREQ_N[6] | |
| 100 | BK58 | PCIE_CLKREQ_N[7] | |
| 101 | BL56 | UART0_RXD | |
| 102 | CA52 | UART0_TXD | |
| 103 | BH54 | SMB_LAN_CLK | |
| 104 | BK54 | SMB_LAN_DATA | |
| 105 | CA50 | ERROR_N[2] | |
| 106 | BR58 | ERROR_N[1] | |
| 107 | BL52 | ERROR_N[0] | |
| 108 | BW50 | IERR_N | |
| 109 | BN52 | MCERR_N | |
| 110 | BY51 | SMB_LEG_CLK | |
| 111 | BV49 | SMB_LEG_DATA | |

## Table 26-1. GPIO Ball and Pad Name (Sheet 4 of 5)

| Number | Ball/Pin | Ball/Pin Name | Notes |
|---|---|---|---|
| 112 | BW48 | SMB_LEG_ALRT_N | |
| 113 | BK50 | SMB_HOST_DATA | |
| 114 | BR50 | SMB_HOST_CLK | |
| 115 | BY49 | SMB_PECI_DATA | |
| 116 | BR54 | SMB_PECI_CLK | |
| 117 | BV47 | SMB_ME_SMT0_DATA | |
| 118 | BR46 | SMB_ME_SMT0_CLK | |
| 119 | BG52 | SMB_ME_SMT0_ALRT_N | |
| 120 | BD54 | USB_OC_N | |
| 121 | BG48 | FLEX_CLK_SE[0] | |
| 122 | CA48 | FLEX_CLK_SE[1] | |
| 123 | BN48 | EMMC_CMD | |
| 124 | BL44 | EMMC_STROBE | |
| 125 | BK46 | EMMC_CLK | |
| 126 | BW45 | EMMC_D[0] | |
| 127 | BV44 | EMMC_D[1] | |
| 128 | BY44 | EMMC_D[2] | |
| 129 | CA45 | EMMC_D[3] | |
| 130 | BU43 | EMMC_D[4] | |
| 131 | CA43 | EMMC_D[5] | |
| 132 | BY42 | EMMC_D[6] | |
| 133 | BV42 | EMMC_D[7] | |
| 134 | AY52 | DFX_PORT[9] | DFX Set2 Pads |
| 135 | BR42 | DFX_PORT[10] | |
| 136 | BK42 | DFX_PORT[11] | |
| 137 | BW41 | DFX_PORT[12] | |
| 138 | BE48 | DFX_PORT[13] | |
| 139 | CA41 | DFX_PORT[14] | |
| 140 | BV40 | DFX_PORT[15] | |

## Table 26-1. GPIO Ball and Pad Name (Sheet 5 of 5)

| Number | Ball/Pin | Ball/Pin Name | Notes |
|---|---|---|---|
| 141 | N/A | N/A | DFX Set1 Pads |
| 142 | N/A | N/A | |
| 143 | N/A | N/A | |
| 144 | N/A | N/A | |
| 145 | N/A | N/A | |
| 146 | BY38 | CX_PRDY_N | |
| 147 | BH42 | CX_PREQ_N | |
| 148 | BE44 | CTBTRIGINOUT | |
| 149 | CA39 | CTBTRIGOUT | |
| 150 | BM38 | GPIO[150] | |
| 151 | BW37 | GPIO[151] | |
| 152 | CA37 | GPIO[152] | |

***Note:*** There are four types of 1.05V pads:
- DFX Set1 pads
- DFX Set2 pads
- SVID pads
- Thermal Management (TM) pads

The 1.05V pads have different buffer impedance ranges.

## 26.2 Pad Description

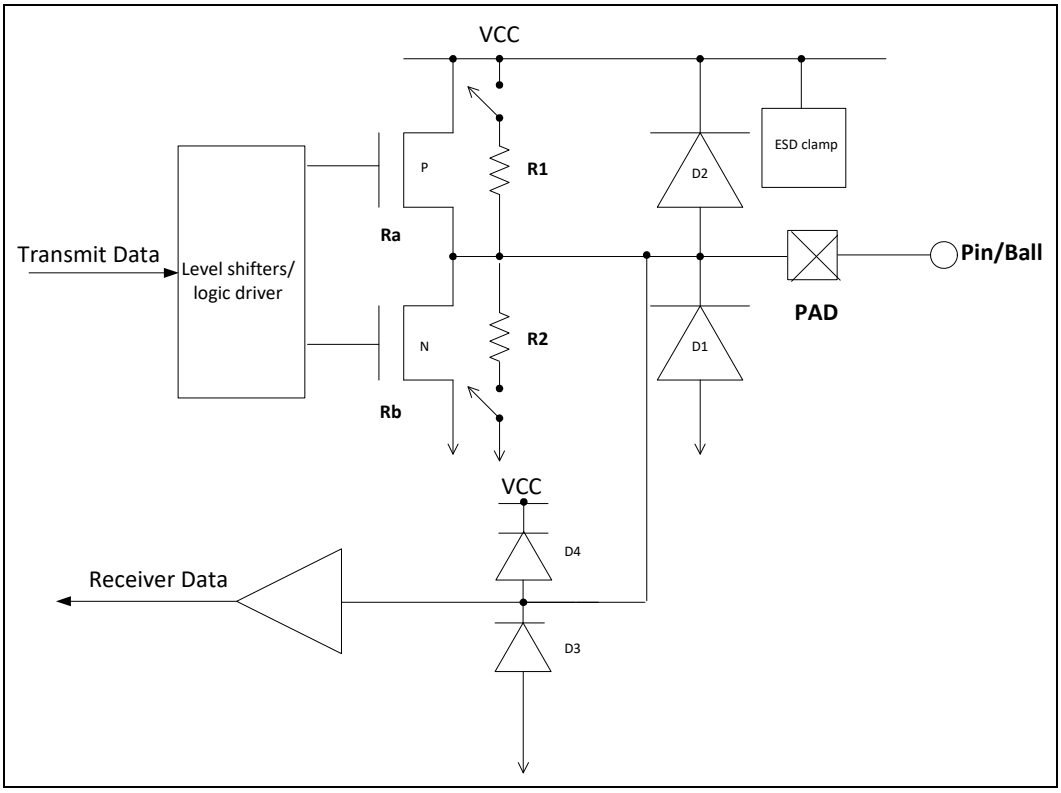
When the signal function is a native function signal (not a GPIO function) the signal is set by the SoC at reset time to be default values. Later on, once BIOS starts, the BIOS can keep all the defaults or alter settings, and then lock all of the configuration registers to be RO (Read Only).

When the signal function is a GPIO mode, the shared-signal pad circuitry contains specific, internal controls including individual pin signal multiplexing, pull-up/pull-down values, pin states and output buffer type (e.g., open drain). See Figure 26-2 for internal pad circuitry. Not all GPIO modes support the pull-up, pull-down options (R1 and R2) shown in Figure 26-2.

See the PAD_CFG_DW1 register configuration options for each GPIO for details of what is supported.

Some of the shared pads are also hard pin straps and have additional pull-up/pull-down circuitry. See Section 4.1, "SoC Hard (Pin) Straps" on page 193.

**Figure 26-2. Custom GPIO Circuit Definition**




*Note:* The internal pull-up resistor option is not supported for 1.8V GPIOs. The internal pull-down resistor option is not supported for 3.3V GPIOs.

The interface power source (VCC) for each pad is fixed and can be one of the three voltage levels (1.05V, 1.8V, or 3.3V).

## 26.3 Signal Configuration Overview

Other than the signal pins mentioned in Chapter 10, "Flexible I/O Adapter (FIA) Overview", the SoC contains signal pins that are shared by more than one native function. They can also be programmed to be Customer GPIO signals.

All 152 of these pins power-up in the high-impedance state and remain in that state until the internal reset for the pin circuitry deasserts. The pins are reset to the default state during a Warm Reset or a Cold Reset (See Chapter 33, "Resets and Voltage Sequencing"). The shared-signal pins that default as output signals are designed to be glitch-free during power-up and reset. For shared-signal pins that by default are Customer GPIO input pins, an external pull-up or pull-down resistor must be used to keep the SoC receiver from triggering an interrupt or event during the internal reset and when the reset is de-asserted.

In many cases, the customer platform design requires an SoC native signal that shares a pin with other SoC native signals. If the desired signal has a Function number that is not the one designated as "Default" in the "Pin States" column of Table 39-1, "Signal Pin Name Directory - Sorted by Name" on page 995, the Pad Mode [2:0] bit field of the pin Pad Configuration DW0 control register (PAD_CFG_DW0) can be programmed to alter the Function number to the desired signal.

If none of the shared SoC native signals are needed by the platform, the customer can set the PAD_CFG_DW0 to Function = 0 (Pad Mode [2:0] = 000), which is always a "Customer GPIO" function.

For pins, the PAD_CFG_DW0 control register and the Pad Configuration DW1 control register (PAD_CFG_DW1), contain bit fields that are customer programmable; used to create the desired signal driver, receiver, and internal pull-up or pull-down termination. Some Customer GPIOs generate interrupts and trigger events.

### 26.3.1 Pad Ownership

The Pad Ownership registers designate whether the Host, Intel® ME, or IE has control of the pad. The Pad Ownership Register (PAD_OWN) uses a two bit code to indicate the ownership of pad.

00 = Host GPIO Mode
01 = Intel® ME GPIO Mode
10 = Reserved
11 = IE GPIO Mode

When a Customer GPIO is programmed to generate an interrupt to the PIC or IOxAPIC, the GPIO Host Mode is also set to either Host GPIO ACPI Mode or GPIO Driver Mode. This is indicated by the Host Software Pad Ownership (HOSTSW_OWN) Register 1-bit code.

0 = GPIO ACPI Mode
1 = GPIO Driver Mode

*Note:*

1. All detail description about the Pad in this chapter is focused on Host GPIO Mode.
2. When the Pad is set as ME GPIO Mode, Intel® ME owns the Pad and controls the Pad. No customization is doable from the customer point of view.
3. When the Pad is set as IE GPIO Mode, the IE owns the Pad.

The BIOS default code does not allow all GPIOs to be reassigned to an IE. Most pins have a native function that prevents assignment to IE as a true GPIO. Please see Table 26-2 for details.

**Table 26-2. Pin Name Descriptions**

| Pin Name | Usage Description |
|---|---|
| GPIO[3] | No native functions assigned; pure GPIO only |
| GPIO[5], GPIO[6], GPIO[7] | Native functions all belong to IE; IE can tradeoff internal features vs pure GPIO |
| SMB4_CSME0_DATA, SMB4_CSME0_CLK | Native functions shared by IE and ME; can tradeoff their relative features vs pure GPIO |
| PCIE_CLKREQ0_N, PCIE_CLKREQ1_N, PCIE_CLKREQ2_N, PCIE_CLKREQ3_N | Native functions shared with PCIe CLKREQ—can allocate some for IE GPIO if platform doesn't require so many CLKREQ |
| GPIO[10], GPIO[11] | Available |
| GPIO[12] | Available if not using TPM |
| GPIO[0], GPIO[1], GPIO[2], GPIO[4], GPIO[8], GPIO[9] | Can be GPIO by IE by turning off the features |

***Note:*** For ME recovery mode, SPS FW is able to use any SoC GPIO pin to configure Recover_Jumper mode by using Intel SPS Flash Image Tool (spsFITc.exe). For guidance of configuration, please refer to Tools for Intel Server Platform Service Firmware 4.0 User Guide in SPS release package.

### 26.3.2 Pad Multiplex Function

The Pad is set as GPIO mode (GPIO ACPI Mode / GPIO Driver Mode) or native function mode, through the Pad Configuration DW0 (PAD_CFG_DW0) register bit [12:10] (Pad Mode).

- 000 = GPIO Controller controls the Pad
- 001 = Native Function 1
- 010 = Native Function 2
- 011 = Native Function 3
- 100 = Native Function 4
- ...through
- 111 = Native Function 7

When the PAD is dedicated GPIO only, the PAD_CFG-DW0 bit [12:10] is all 000.

When the Pad Mode (PMode) is 0, the GPIO controller has the ability to control the RX/TX Disable and RX/TX State through the PAD_CFG_DW0 register bit [9:8] (GPIO RX/TX Disable) and bit[1:0] (GPIO RX/TX State). The pad is either an input or an output signal with different configurations.

When the PMode is not equal to 0, the corresponding GPIO works as the native function mode. The default value of the Pad Mode (PMode) field, also called the signal function value, is assigned when the SoC is reset.

### 26.3.3 Pad Internal Pull-up / Pull-down Configure

The pad has internal weak pull up/down termination which is controlled by register PAD_CFG_DW1 bit [13:10] (TERM). It is expected software/FW/BIOS shall program the termination settings appropriately. Some common sequences are:

1. Termination settings must be programmed before changing a pad from output to input to avoid the pad state “floating” momentarily.
2. If a given pad is configured as an output, termination must be enabled until the TXDATA is driven properly, to avoid pad state “floating”. This means the TX enable configuration has to be enabled in order to enable the TX buffer before termination is released.

## 26.3.4 Pad Configuration Logical Control Summary

The pad is configured as an input signal or an output signal, including an open drain output signal. When the signal is set as a Customer GPIO function Table 26-3 through Table 26-5 show how to configure the signal to be input or output.

**Figure 26-3. GPIO Logic Overview Diagram**

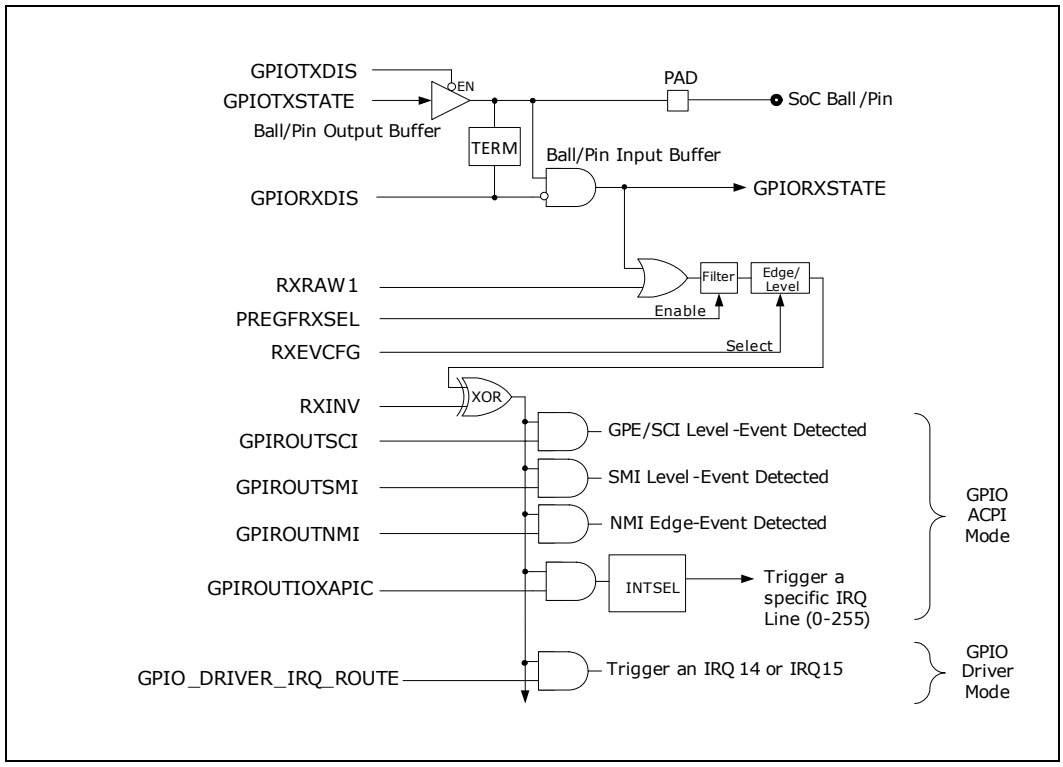



**Table 26-3. Input Signal Configuration**

| Pad | GPIORXDIS | GPIOTXDIS | TERM | GPIORXSTATE | GPIOTXSTATE |
|---|---|---|---|---|---|
| **0** | 0 | 1 | 0000 | 0 | X |
| **1** | 0 | 1 | 0000 | 1 | X |

**Table 26-4. Output Signal Configuration**

| Pad | GPIORXDIS | GPIOTXDIS | TERM | GPIORXSTATE | GPIOTXSTATE |
|---|---|---|---|---|---|
| **0*** | 1 | 0 | 0000 | 0 | 0 |
| **1*** | 1 | 0 | 0000 | 0 | 1 |
| **Hi-Z*** | 1 | 1 | 0000 | 0 | 0 |
| **Hi-Z** | 1 | 0 | 0000 | 0 | 0 |
| **0** | 1 | 1 | 0010/0100 | 0 | 0 |
| **1** | 1 | 1 | 1010/1100 | 0 | 0 |
| * Open Drain Output Signal Configuration. | | | | | |

### 26.3.5 Non-Maskable Interrupt (NMI)

When a Customer GPIO is customer-configured to be an input, an NMI can be generated by following programming steps:

1. Set Host Ownership - PAD_OWN[1:0] = '00'.
2. Set GPIO ACPI Mode - HOSTSW_OWN = '0'.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. NMI Routing enabling - PAD_CFG_DW0 bit [17] (GPIRoutNMI) = '1'.
5. GPI NMI Enable - GPI_NMI_EN = '1'.
6. GPI NMI Status - GPI_NMI_STS= '1'.

*Note:* Some GPIO signals do not support NMI. Please see detail in Table 26-5, "GPIO Pad Configuration Description".

### 26.3.6 System Management Interrupt (SMI)

When a Customer GPIO is customer-configured to be an input, an SMI can be generated by following programming steps:

1. Set Host Ownership - PAD_OWN[1:0]= '00'.
2. Set GPIO ACPI Mode - HOSTSW_OWN = '0'.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. SMI Routing enabling - PAD_CFG_DW0 bit [18] (GPIRoutSMI) = '1'.
5. GPI SMI Enable - GPI_SMI_EN = '1'.
6. GPI SMI Status - GPI_SMI_STS= '1'.

*Note:* Some GPIO signals do not support SMI. Please see detail in Table 26-5, "GPIO Pad Configuration Description".

### 26.3.7 System Control Interrupt (SCI)/General Purpose Event (GPE)

When a Customer GPIO is customer-configured to be an input, an SCI can be generated by following programming steps:

1. Set Host Ownership - PAD_OWN[1:0]= '00'.
2. Set GPIO ACPI Mode - HOSTSW_OWN = '0'.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. SCI/GPE Routing enabling - PAD_CFG_ DW0 bit [19] (GPIRoutSCI) = '1'.
5. GPI General Purpose Events Enable - GPI_GPE_EN = '1'.
6. GPI General Purpose Events Status - GPI_GPE_STS = '1'.

### 26.3.8 GPIO to IOxAPIC

When a Customer GPIO is customer-configured to be an input, a GPIO can directly send an interrupt to the IOxAPIC by following programming steps:

1. Set Host Ownership - PAD_OWN[1:0]= '00'.
2. Set GPIO ACPI Mode - HOSTSW_OWN = '0'.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. IOxAPIC Routing enabling - PAD_CFG_DW0 bit [20] (GPIRoutIOxAPIC) = '1'.
5. Specific IRQ Line Enable - PAD_CFG_DW1 bits [6:0] Interrupt Select (IntSel) = '0-119'.

*Note:* Each GPIO signal has a fixed Interrupt line. Please see detail in Table 27-8.

### 26.3.9 Legacy Interrupt Request (IRQ)

When a Customer GPIO is customer-configured to be an input, a Legacy IRQ can be generated by following programming steps:

1. Set Host Ownership - PAD_OWN[1:0]= '00'.
2. Set GPIO Driver Mode - HOSTSW_OWN = '1'.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. Miscellaneous Configuration Register - MISCCFG bit [3] (GPIO_DRIVER_IRQ_ROUTE) = '0' (IRQ14) or '1' (IRQ15).
5. GPI Interrupt Enable - GPI_INT_EN = '1'.
6. GPI Interrupt Status - GPI_INT_STS= '1'.

### 26.3.10 Pad Level/Edge Sensitive Detect Enable

The interrupt or wake triggering event for the input Customer GPIO are delivered to the GPIO controller as either Edge detected or Level detected by following program steps:

1. Set Host Ownership - PAD_OWN[1:0]= '00'.
2. Set GPIO Mode based on interrupt type.
3. Set GPIO Input Mode – PAD_CFG_DW0 [12:10] = '000'.
4. Select the polarity inversion stage - PAD_CFG_DW0 bit [23] (RXINV)
5. Enable RX Level/Edge configuration - PAD_CFG_DW0 two bits [26:25] (RXEVCFG).

The default uses Edge detection logic with no inversion.

## 26.4 Registers Lock Down

The intended use for the pad Configuration Lock (PADCFGLOCK), is for BIOS to lock down the registers related to the pad configuration. This is one-bit-per-pad to unlock/lock certain pad specific fields. After locking, the specific fields become Read-Only. BIOS can unlock PADCFGLOCK by sending opcode 0x13 to the individual GPIO community.

When PADCFGLOCK is written from "1" to "0" (unlock), a synchronous SMI# is generated when enabled. When the unlock is enabled, it cannot be disabled until PMU_PLRTST_N happens. This ensures that only the SMM code changes the register field settings after they are locked.

This configuration is meaningful and affects only the pad owned by the host - PAD_OWN[1:0]= '00'.

## 26.5 MMIO Address Access

All GPIO registers must be accessed through the Memory Mapped I/O (MMIO) space with the address offsets to SBREG_BAR (B:0 D:31 F:1).

The pad Control Register and Group Register field names identify the specific GPIO by its Pin Name as shown in Table 39-1, "Signal Pin Name Directory - Sorted by Name" on page 995. The Pin Name is used as the suffix of the register name. For example, the register PAD_CFG_DW0_GPIO_0 and the bit PAD_OWN_GPIO_0 for the pin named "*_GPIO_0" has part of the register name.

*Note:* The register name with the suffix shown in Table 26-5 is used by BIOS/FW for code development/register programming. Some register names have a different suffix name than the name of the pin. These are noted in Table 26-5.

**Table 26-5. GPIO Pad Configuration Description (Sheet 1 of 5)**

| GPIO | Suffix of Pad Register Name | Pad Configuration DW0 Offset (hex) | Pad Configuration DW1 Offset (hex) | SMI/NMI Capability | IOxAPIC Interrupt INTSEL (hex) | Comment |
|---|---|---|---|---|---|---|
| 14 | *_GBE0_SDP0 | C2_0400 | C2_0404 | | 18 | 1 |
| 15 | *_GBE1_SDP0 | C2_0408 | C2_040C | | 19 | 1 |
| 16 | *_GBE0_SDP1 | C2_0410 | C2_0414 | | 1A | 1 |
| 17 | *_GBE1_SDP1 | C2_0418 | C2_041C | | 1B | 1 |
| 18 | *_GBE0_SDP2 | C2_0420 | C2_0424 | | 1C | 1, 2 |
| 19 | *_GBE1_SDP2 | C2_0428 | C2_042C | | 1D | 1 |
| 20 | *_GBE0_SDP3 | C2_0430 | C2_0434 | | 1E | 1, 2 |
| 21 | *_GBE1_SDP3 | C2_0438 | C2_043C | | 1F | 1, 2 |
| 22 | *_GBE2_LED0 | C2_0440 | C2_0444 | | 20 | 1 |
| 23 | *_GBE2_LED1 | C2_0448 | C2_044C | | 21 | 1 |
| 24 | *_GBE0_I2C_CLK | C2_0450 | C2_0454 | | 22 | 1 |
| 25 | *_GBE0_I2C_DATA | C2_0458 | C2_045C | | 23 | 1, 2 |
| 26 | *_GBE1_I2C_CLK | C2_0460 | C2_0464 | | 24 | 1 |
| 27 | *_GBE1_I2C_DATA | C2_0468 | C2_046C | | 25 | 1, 2 |
| 28 | *_NCSI_RXD0 | C2_0470 | C2_0474 | | 26 | |

1. The Ball/Pin Name is not same as the suffix of the pad's register name.
2. Until RSMRST_N de-asserts these signals are indeterminate.

## Table 26-5. GPIO Pad Configuration Description (Sheet 2 of 5)

| GPIO | Suffix of Pad Register Name | Pad Configuration DW0 Offset (hex) | Pad Configuration DW1 Offset (hex) | SMI/NMI Capability | IOxAPIC Interrupt INTSEL (hex) | Comment |
|---|---|---|---|---|---|---|
| 29 | *_NCSI_CLK_IN | C2_0478 | C2_047C | | 27 | |
| 30 | *_NCSI_RXD1 | C2_0480 | C2_0484 | | 28 | |
| 31 | *_NCSI_CRS_DV | C2_0488 | C2_048C | | 29 | |
| 32 | *_NCSI_ARB_IN | C2_0490 | C2_0494 | | 2A | 2 |
| 33 | *_NCSI_TX_EN | C2_0498 | C2_049C | | 2B | |
| 34 | *_NCSI_TXD0 | C2_04A0 | C2_04A4 | | 2C | 2 |
| 35 | *_NCSI_TXD1 | C2_04A8 | C2_04AC | | 2D | |
| 36 | *_NCSI_ARB_OUT | C2_04B0 | C2_04B4 | | 2E | 2 |
| 37 | *_GBE0_LED0 | C2_04B8 | C2_04BC | | 2F | 1 |
| 38 | *_GBE0_LED1 | C2_04C0 | C2_04C4 | | 30 | 1 |
| 39 | *_GBE1_LED0 | C2_04C8 | C2_04CC | | 31 | 1 |
| 40 | *_GBE1_LED1 | C2_04D0 | C2_04D4 | | 32 | 1 |
| 0 | *_GPIO_0 | C2_04D8 | C2_04DC | Yes | 33 | |
| 41 | *_PCIE_CLKREQ0_N | C2_04E0 | C2_04E4 | | 34 | 2 |
| 42 | *_PCIE_CLKREQ1_N | C2_04E8 | C2_04EC | | 35 | |
| 43 | *_PCIE_CLKREQ2_N | C2_04F0 | C2_04F4 | | 36 | |
| 44 | *_PCIE_CLKREQ3_N | C2_04F8 | C2_04FC | | 37 | |
| 45 | *_PCIE_CLKREQ4_N | C2_0500 | C2_0504 | | 38 | 2 |
| 1 | *_GPIO_1 | C2_0508 | C2_050C | Yes | 39 | 2 |
| 2 | *_GPIO_2 | C2_0510 | C2_0514 | Yes | 3A | 2 |
| 47 | *_SVID_ALERT_N | C2_0518 | C2_051C | | 3B | |
| 48 | *_SVID_DATA | C2_0520 | C2_0524 | | 3C | |
| 49 | *_SVID_CLK | C2_0528 | C2_052C | | 3D | |
| 50 | *_THERMTRIP_N | C2_0530 | C2_0534 | | 3E | |
| 51 | *_PROCHOT_N | C2_0538 | C2_053C | | 3F | |
| 52 | *_MEMHOT_N | C2_0540 | C2_0544 | | 40 | |
| 53 | *_DFX_PORT_CLK0 | C5_0400 | C5_0404 | | 41 | |
| 54 | *_DFX_PORT_CLK1 | C5_0408 | C5_040C | | 42 | |
| 55 | *_DFX_PORT0 | C5_0410 | C5_0414 | | 43 | |
| 56 | *_DFX_PORT1 | C5_0418 | C5_041C | | 44 | |
| 57 | *_DFX_PORT2 | C5_0420 | C5_0424 | | 45 | |
| 58 | *_DFX_PORT3 | C5_0428 | C5_042C | | 46 | |
| 59 | *_DFX_PORT4 | C5_0430 | C5_0434 | | 47 | |
| 60 | *_DFX_PORT5 | C5_0438 | C5_043C | | 48 | |
| 61 | *_DFX_PORT6 | C5_0440 | C5_0444 | | 49 | |
| 62 | *_DFX_PORT7 | C5_0448 | C5_044C | | 4A | |
| 63 | *_DFX_PORT8 | C5_0450 | C5_0454 | | 4B | |

1. The Ball/Pin Name is not same as the suffix of the pad's register name.
2. Until RSMRST_N de-asserts these signals are indeterminate.

## Table 26-5. GPIO Pad Configuration Description (Sheet 3 of 5)

| GPIO | Suffix of Pad Register Name | Pad Configuration DW0 Offset (hex) | Pad Configuration DW1 Offset (hex) | SMI/NMI Capability | IOxAPIC Interrupt INTSEL (hex) | Comment |
|---|---|---|---|---|---|---|
| 134 | *_DFX_PORT9 | C5_0458 | C5_045C | | 4C | |
| 135 | *_DFX_PORT10 | C5_0460 | C5_0464 | | 4D | |
| 136 | *_DFX_PORT11 | C5_0468 | C5_046C | | 4E | |
| 137 | *_DFX_PORT12 | C5_0470 | C5_0474 | | 4F | |
| 138 | *_DFX_PORT13 | C5_0478 | C5_047C | | 50 | |
| 139 | *_DFX_PORT14 | C5_0480 | C5_0484 | | 51 | |
| 140 | *_DFX_PORT15 | C5_0488 | C5_048C | | 52 | |
| 12 | *_GPIO_12 | C5_0490 | C5_0494 | Yes | 53 | |
| 13 | *_SMB5_GBE_ALRT_N | C5_0498 | C5_049C | Yes | 54 | 1 |
| 98 | *_PCIE_CLKREQ5_N | C5_04A0 | C5_04A4 | Yes | 55 | |
| 99 | *_PCIE_CLKREQ6_N | C5_04A8 | C5_04AC | | 56 | |
| 100 | *_PCIE_CLKREQ7_N | C5_04B0 | C5_04B4 | | 57 | 2 |
| 101 | *_UART0_RXD | C5_04B8 | C5_04BC | Yes | 58 | 2 |
| 102 | *_UART0_TXD | C5_04C0 | C5_04C4 | Yes | 59 | 2 |
| 103 | *_SMB5_GBE_CLK | C5_04C8 | C5_04CC | Yes | 5A | 1, 2 |
| 104 | *_SMB5_GBE_DATA | C5_04D0 | C5_04D4 | | 5B | 1 |
| 105 | *_ERROR2_N | C5_04D8 | C5_04DC | | 5C | 2 |
| 106 | *_ERROR1_N | C5_04E0 | C5_04E4 | | 5D | 2 |
| 107 | *_ERROR0_N | C5_04E8 | C5_04EC | | 5E | 2 |
| 108 | *_IERR_N | C5_04F0 | C5_04F4 | | 5F | |
| 109 | *_MCERR_N | C5_04F8 | C5_04FC | | 60 | 2 |
| 110 | *_SMB0_LEG_CLK | C5_0500 | C5_0504 | | 61 | |
| 111 | *_SMB0_LEG_DATA | C5_0508 | C5_050C | | 62 | |
| 112 | *_SMB0_LEG_ALRT_N | C5_0510 | C5_0514 | | 63 | |
| 113 | *_SMB1_HOST_DATA | C5_0518 | C5_051C | | 64 | 2 |
| 114 | *_SMB1_HOST_CLK | C5_0520 | C5_0524 | | 65 | |
| 115 | *_SMB2_PECI_DATA | C5_0528 | C5_052C | | 66 | 2 |
| 116 | *_SMB2_PECI_CLK | C5_0530 | C5_0534 | | 67 | |
| 117 | *_SMB4_CSME0_DATA | C5_0538 | C5_053C | | 68 | |
| 118 | *_SMB4_CSME0_CLK | C5_0540 | C5_0544 | | 69 | |
| 119 | *_SMB4_CSME0_ALRT_N | C5_0548 | C5_054C | | 6A | |
| 120 | *_USB_OC0_N | C5_0550 | C5_0554 | | 6B | 2 |
| 121 | *_FLEX_CLK_SE0 | C5_0558 | C5_055C | Yes | 6C | |
| 122 | *_FLEX_CLK_SE1 | C5_0560 | C5_0564 | Yes | 6D | 2 |
| 4 | *_GPIO_4 | C5_0568 | C5_056C | | 6E | |
| 5 | *_GPIO_5 | C5_0570 | C5_0574 | | 6F | |
| 6 | *_GPIO_6 | C5_0578 | C5_057C | | 70 | |

1. The Ball/Pin Name is not same as the suffix of the pad's register name.
2. Until RSMRST_N de-asserts these signals are indeterminate.

## Table 26-5. GPIO Pad Configuration Description (Sheet 4 of 5)

| GPIO | Suffix of Pad Register Name | Pad Configuration DW0 Offset (hex) | Pad Configuration DW1 Offset (hex) | SMI/NMI Capability | IOxAPIC Interrupt INTSEL (hex) | Comment |
|---|---|---|---|---|---|---|
| 7 | *_GPIO_7 | C5_0580 | C5_0584 | | 71 | |
| 90 | *_SATA0_LED_N | C5_0588 | C5_058C | Yes | 72 | |
| 91 | *_SATA1_LED_N | C5_0590 | C5_0594 | Yes | 73 | |
| 92 | *_SATA_PDETECT0 | C5_0598 | C5_059C | Yes | 74 | |
| 93 | *_SATA_PDETECT1 | C5_05A0 | C5_05A4 | Yes | 75 | |
| 94 | *_SATA0_SDOUT | C5_05A8 | C5_05AC | Yes | 76 | |
| 95 | *_SATA1_SDOUT | C5_05B0 | C5_05B4 | Yes | 18 | |
| 96 | *_UART1_RXD | C5_05B8 | C5_05BC | Yes | 19 | |
| 97 | *_UART1_TXD | C5_05C0 | C5_05C4 | Yes | 1A | 2 |
| 8 | *_GPIO_8 | C5_05C8 | C5_05CC | Yes | 1B | |
| 9 | *_GPIO_9 | C5_05D0 | C5_05D4 | Yes | 1C | 2 |
| 146 | *_CX_PRDY_N | C5_0600 | C5_0604 | | 22 | |
| 147 | *_CX_PREQ_N | C5_0608 | C5_060C | | 23 | |
| 148 | *_CTBTRIGINOUT | C5_0610 | C5_0614 | | 24 | |
| 149 | *_CTBTRIGOUT | C5_0618 | C5_061C | | 25 | |
| 150 | *_DFX_SPARE2 | C5_0620 | C5_0624 | | 26 | 1 |
| 151 | *_DFX_SPARE3 | C5_0628 | C5_062C | | 27 | 1 |
| 152 | *_DFX_SPARE4 | C5_0630 | C5_0634 | | 28 | 1 |
| 79 | *_SUSPWRDNACK | C5_0638 | C5_063C | | 29 | |
| 80 | *_PMU_SUSCLK | C5_0640 | C5_0644 | | 2A | |
| 81 | *_ADR_TRIGGER | C5_0648 | C5_064C | Yes | 2B | |
| 82 | *_PMU_SLP_S45 | C5_0650 | C5_0654 | | 2C | |
| 83 | *_PMU_SLP_S3 | C5_0658 | C5_065C | | 2D | |
| 84 | *_PMU_WAKE_N | C5_0660 | C5_0664 | | 2E | 2 |
| 85 | *_PMU_PWRBTN_N | C5_0668 | C5_066C | | 2F | 2 |
| 86 | *_PMU_RESETBUTTON_N | C5_0670 | C5_0674 | | 30 | 2 |
| 87 | *_PMU_PLTRST_N | C5_0678 | C5_067C | | 31 | 2 |
| 88 | *_SUS_STAT_N | C5_0680 | C5_0684 | | 32 | |
| 89 | *_SLP_S0IX_N | C5_0688 | C5_068C | | 33 | 1 |
| 72 | *_SPI_CS0_N | C5_0690 | C5_0694 | | 34 | |
| 73 | *_SPI_CS1_N | C5_0698 | C5_069C | | 35 | |
| 74 | *_SPI_MOSI_IO0 | C5_06A0 | C5_06A4 | | 36 | |
| 75 | *_SPI_MOSI_IO1 | C5_06A8 | C5_06AC | | 37 | |
| 76 | *_SPI_IO2 | C5_06B0 | C5_06B4 | | 38 | 2 |
| 77 | *_SPI_IO3 | C5_06B8 | C5_06BC | | 39 | 2 |
| 78 | *_SPI_CLK | C5_06C0 | C5_06C4 | | 3A | 2 |
| 64 | *_ESPI_IO0 | C5_06D0 | C5_06D4 | | 3C | |

1. The Ball/Pin Name is not same as the suffix of the pad's register name.
2. Until RSMRST_N de-asserts these signals are indeterminate.

## Table 26-5. GPIO Pad Configuration Description (Sheet 5 of 5)

| GPIO | Suffix of Pad Register Name | Pad Configuration DW0 Offset (hex) | Pad Configuration DW1 Offset (hex) | SMI/NMI Capability | IOxAPIC Interrupt INTSEL (hex) | Comment |
|---|---|---|---|---|---|---|
| 65 | *_ESPI_IO1 | C5_06D8 | C5_06DC | | 3D | |
| 66 | *_ESPI_IO2 | C5_06E0 | C5_06E4 | | 3E | |
| 67 | *_ESPI_IO3 | C5_06E8 | C5_06EC | | 3F | |
| 68 | *_ESPI_CS0_N | C5_06F0 | C5_06F4 | | 40 | |
| 69 | *_ESPI_CLK | C5_06F8 | C5_06FC | | 41 | |
| 70 | *_ESPI_RST_N | C5_0700 | C5_0704 | | 42 | |
| 71 | *_ESPI_ALRT0_N | C5_0708 | C5_070C | Yes | 43 | |
| 10 | *_GPIO_10 | C5_0710 | C5_0714 | Yes | 44 | |
| 11 | *_GPIO_11 | C5_0718 | C5_071C | Yes | 45 | |
| 123 | *_EMMC_CMD | C5_0728 | C5_072C | Yes | 47 | |
| 124 | *_EMMC_STROBE | C5_0730 | C5_0734 | Yes | 48 | |
| 125 | *_EMMC_CLK | C5_0738 | C5_073C | Yes | 49 | |
| 126 | *_EMMC_D0 | C5_0740 | C5_0744 | Yes | 4A | |
| 127 | *_EMMC_D1 | C5_0748 | C5_074C | Yes | 4B | |
| 128 | *_EMMC_D2 | C5_0750 | C5_0754 | Yes | 4C | |
| 129 | *_EMMC_D3 | C5_0758 | C5_075C | Yes | 4D | |
| 130 | *_EMMC_D4 | C5_0760 | C5_0764 | Yes | 4E | |
| 131 | *_EMMC_D5 | C5_0768 | C5_076C | Yes | 4F | |
| 132 | *_EMMC_D6 | C5_0770 | C5_0774 | Yes | 50 | |
| 133 | *_EMMC_D7 | C5_0778 | C5_077C | Yes | 51 | |
| 3 | *_GPIO_3 | C5_0780 | C5_0784 | Yes | 52 | |

1. The Ball/Pin Name is not same as the suffix of the pad's register name.
2. Until RSMRST_N de-asserts these signals are indeterminate.

For GPIO registers access, there are two ways. One way is using P2SB  to access GPIO registers. Another way is to reach GPIO by PMC through Sideband.

The GPIO controllers have 7 Dwords of GPE status and enable registers (NORTH_ALL_1/0 and SOURTH_DFX/GROUP0/1). But PMC exposes only 3 Dwords to SW. So BIOS must choose which 3 of the 7 to make visible.

Programming steps:

1. BIOS programs GPIO_CFG.GPE0_DW0, GPE0_DW1, GPE0_DW2 with the 3 GPIO groups to make visible. There are now 96 GPE available:
   - GPE[31:0] → GPIO selected by GPE0_DW0
   - GPE[63:32] → GPIO selected by GPE0_DW1
   - GPE[95:64] → GPIO selected by GPE0_DW2

2. BIOS enables GPE within the 96 available by writing to PMC. PMC forwards the writes to the GPIO based on mapping in 1)
   - Program GPE0_EN_31_0 → GPIO selected by GPE0_DW0
   - Program GPE0_EN_63_32 → GPIO selected by GPE0_DW1
   - Program GPE0_EN_95_64 → GPIO selected by GPE0_DW2

3. BIOS checks GPE status within the 96 available by reading from PMC. PMC forwards the reads to the GPIO based on mapping in 1)
   - Read I/O 0x80 → GPIO selected by GPE0_DW0
   - Read I/O 0x84 → GPIO selected by GPE0_DW1
   - Read I/O 0x88 → GPIO selected by GPE0_DW2

### Table 26-6. PMC pci_cfg.PWRMBASE

<table>
<tr><th>MEM Address</th><th>Register</th><th>Field</th><th></th></tr>
<tr><td>0x120</td><td>GPIO_CFG</td><td>GPE0_DWG</td><td>Assigns a specific GPIO register to ABASE.0x80 and ABASE.0x90
<table>
<tr><th>Value</th><th>0x80 maps to this GPIO register</th><th>0x90 maps to this GPIO register</th></tr>
<tr><td>0</td><td>GPI_GPE_STS_NORTH_ALL_0</td><td>GPI_GPE_EN_NORTH_ALL_0</td></tr>
<tr><td>1</td><td>GPI_GPE_STS_NORTH_ALL_1</td><td>GPI_GPE_EN_NORTH_ALL_1</td></tr>
<tr><td>2</td><td>GPI_GPE_STS_SOUTH_DFX_0</td><td>GPI_GPE_EN_SOUTH_DFX_0</td></tr>
<tr><td>3</td><td>GPI_GPE_STS_SOUTH_GROUP0_0</td><td>GPI_GPE_EN_SOUTH_GROUP0_0</td></tr>
<tr><td>4</td><td>GPI_GPE_STS_SOUTH_GROUP0_1</td><td>GPI_GPE_EN_SOUTH_GROUP0_1</td></tr>
<tr><td>5</td><td>GPI_GPE_STS_SOUTH_GROUP1_0</td><td>GPI_GPE_EN_SOUTH_GROUP1_0</td></tr>
<tr><td>6</td><td>GPI_GPE_STS_SOUTH_GROUP1_1</td><td>GPI_GPE_EN_SOUTH_GROUP1_1</td></tr>
<tr><td>7</td><td>Reserved</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>GPE0_DW1</td><td>Assigns a specific GPIO register to ABASE.0x84 and ABASE.0x94
<table>
<tr><th>Value</th><th>0x84 maps to this GPIO register</th><th>0x94 maps to this GPIO register</th></tr>
<tr><td>0</td><td>GPI_GPE_STS_NORTH_ALL_0</td><td>GPI_GPE_EN_NORTH_ALL_0</td></tr>
<tr><td>1</td><td>GPI_GPE_STS_NORTH_ALL_1</td><td>GPI_GPE_EN_NORTH_ALL_1</td></tr>
<tr><td>2</td><td>GPI_GPE_STS_SOUTH_DFX_0</td><td>GPI_GPE_EN_SOUTH_DFX_0</td></tr>
<tr><td>3</td><td>GPI_GPE_STS_SOUTH_GROUP0_0</td><td>GPI_GPE_EN_SOUTH_GROUP0_0</td></tr>
<tr><td>4</td><td>GPI_GPE_STS_SOUTH_GROUP0_1</td><td>GPI_GPE_EN_SOUTH_GROUP0_1</td></tr>
<tr><td>5</td><td>GPI_GPE_STS_SOUTH_GROUP1_0</td><td>GPI_GPE_EN_SOUTH_GROUP1_0</td></tr>
<tr><td>6</td><td>GPI_GPE_STS_SOUTH_GROUP1_1</td><td>GPI_GPE_EN_SOUTH_GROUP1_1</td></tr>
<tr><td>7</td><td>Reserved</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td></td><td></td><td>GPE0_DW2</td><td>Assigns a specific GPIO register to ABASE.0x88 and ABASE.0x98
<table>
<tr><th>Value</th><th>0x88 maps to this GPIO register</th><th>0x98 maps to this GPIO register</th></tr>
<tr><td>0</td><td>GPI_GPE_STS_NORTH_ALL_0</td><td>GPI_GPE_EN_NORTH_ALL_0</td></tr>
<tr><td>1</td><td>GPI_GPE_STS_NORTH_ALL_1</td><td>GPI_GPE_EN_NORTH_ALL_1</td></tr>
<tr><td>2</td><td>GPI_GPE_STS_SOUTH_DFX_0</td><td>GPI_GPE_EN_SOUTH_DFX_0</td></tr>
<tr><td>3</td><td>GPI_GPE_STS_SOUTH_GROUP0_0</td><td>GPI_GPE_EN_SOUTH_GROUP0_0</td></tr>
<tr><td>4</td><td>GPI_GPE_STS_SOUTH_GROUP0_1</td><td>GPI_GPE_EN_SOUTH_GROUP0_1</td></tr>
<tr><td>5</td><td>GPI_GPE_STS_SOUTH_GROUP1_0</td><td>GPI_GPE_EN_SOUTH_GROUP1_0</td></tr>
<tr><td>6</td><td>GPI_GPE_STS_SOUTH_GROUP1_1</td><td>GPI_GPE_EN_SOUTH_GROUP1_1</td></tr>
<tr><td>7</td><td>Reserved</td><td>Reserved</td></tr>
</table></td></tr>
</table>

### Table 26-7. PMC pci_cfg.ABASE

| IO Address | Register | |
|---|---|---|
| 0x80 | GPE0_STS_31_0 | R/W is redirected to GPE STATUS register per mapping in GPIO_CFG.GPE0_DW0 |
| 0x84 | GPE0_STS_63_32 | R/W is redirected to GPE STATUS register per mapping in GPIO_CFG.GPE0_DW1 |
| 0x88 | GPE0_STS_95_64 | R/W is redirected to GPE STATUS register per mapping in GPIO_CFG.GPE0_DW2 |
| 0x8C | GPE0_STS_127_96 | STATUS of special GPE events defined by PMC; this register physically lives in PMC |
| 0x90 | GPE0_EN_31_0 | R/W is redirected to GPE ENABLE register per mapping in GPIO_CFG.GPE0_DW0 |
| 0x94 | GPE0_EN_63_32 | R/W is redirected to GPE ENABLE register per mapping in GPIO_CFG.GPE0_DW1 |
| 0x98 | GPE0_EN_95_64 | R/W is redirected to GPE ENABLE register per mapping in GPIO_CFG.GPE0_DW2 |
| 0x9C | GPE0_EN_127_96 | ENABLE of special GPE events defined by PMC; this register physically lives in PMC |

§ §

# 27 System Management Bus (SMBus) Controller - Overview

The SoC provides multiple System Management Bus (SMBus) 2.0 controllers. The interfaces comply to *System Management Bus (SMBus) Specification, Version 3.0*. The controllers provide a mechanism for various system components [SoC, Baseboard Management Controller (BMC), etc.] to communicate with each other and with the entire system. The SoC is also capable of operating in a mode that communicates with $I^2C$*-compatible devices.

*Note:* There are only seven sets of SMBus pins so a maximum of seven SMBus controllers maybe active.

The SoC includes the following SMBus controllers:

- Legacy SMBus (wire ORed with CLTT SMBus):
  - IA cores use the master/slave legacy interface to communicate with managed devices in the system.
  - The BIOS uses this interface to detect and configure memory devices.
- Host SMBus:
  - IA cores use this master/slave interface to enable the SoC to send/receive out-of-band messages over the SMBus to peripherals. The controller operation is DMA-based and utilizes descriptors to initiate transactions on the bus.
- Platform Environment Control Interface (PECI) SMBus:
  - A slave only interface which enables external management devices (BMC) to obtain system information like thermal data through PECI commands encapsulated within SMBus packets to the SoC.
- Three Intel® Management Engine (Intel® ME) SMBuses:
  - The Intel® ME SMBus Message Transport (SMT) controllers are configured as a master or slave for internal agents inside the SoC to communicate to the external management devices [BMC/External Circuitry (EC)] over SMBus.
  - The silicon-enabling firmware and Intel® Node Manager (Intel® NM) of the Intel® Server Platform Services (SPS) firmware defines the SMT interface configuration and connection.
- LAN SMBus:
  - This master/slave sideband interface provides a mechanism for the network management traffic to be routed to and from a management controller (BMC). The SMBus in each LAN controller is wire-ANDed in silicon.
- Three Innovation Engine (IE) SMBuses:
  - IE uses master/slave interfaces to communicate with devices in the system.

**Table 27-1. References**

| Document Title | Document ID / Location |
|---|---|
| *System Management Bus (SMBus) Specification, Version 3.0* | http://smbus.org/specs/SMBus_3_0_20141220.pdf |

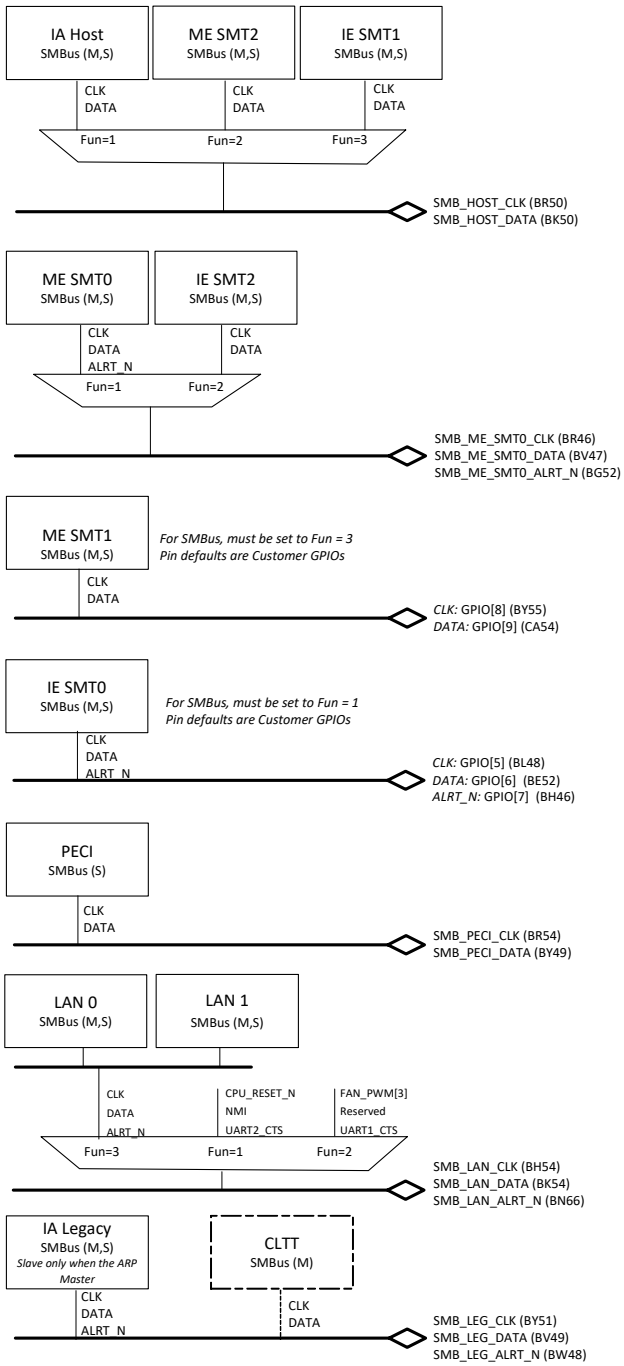
## 27.1 General Architecture

At its network layer, the *System Management Bus (SMBus) Specification, Version 3.0* refers to three types of devices:

- **Slave Device** - A device that is receiving or responding to a command.
- **Master** - A device that issues commands, generates the clocks, and terminates the transfer.
- **Host** - A specialized master that provides the main interface to the system CPU. A host must be a master-slave and must support the SMBus host notify protocol. At most, one host exists in a system.

Figure 27-1 shows the overall SMBus arrangement in the SoC, M = Master, S = Slave.

**Figure 27-1. Arrangement of SMBus in SoC**




***Note:*** If software running on the IA uses the legacy SMBus (SMBus master), then there are multiple masters active. When multiple master are active on the same bus, software using the legacy SMBus controller must check for Bus Error (BERR) in the Status register (HSTS) to determine if a bus collision has occurred, and retry the transaction if needed.

## 27.2 Feature List

The following are key features of the various controllers:

- Legacy SMBus:
  - Serial interface providing a mechanism for the processor to initiate communications with the SMBus peripherals
  - Access initiated through registers in Host Root Memory Space and Host Root I/O Space.
  - The Memory Reference Code (MRC) uses this bus to detect and configure the DIMMs
  - Closed loop thermal throttling (CLTT) which requires the controller to act as a bus master to monitor DIMM temperature (Refer Chapter 35, "Thermal Management")
  - Chapter 28, "SMBus Controller - Legacy"
- Host SMBus:
  - Operates as an SMBus master or slave
  - Supports Address Resolution Protocol (ARP) in master or slave mode
  - Access initiated through the descriptors
  - OS controlled to communicate with any platform device
  - Packet Error Checking (PEC) is enabled for SMBus transactions
  - Chapter 29, "SMBus Controller - Host"
- PECI SMBus:
  - Supports external BMC
  - Operates in slave mode only
  - Supports one command at a time
  - Chapter 30, "SMBus Controller - Platform Environment Control Interface"
- Intel® ME SMBuses:
  - Sends Out-Of-Band (OOB) manageability messages to managed devices or management controllers
  - SMT controllers operate with full master and slave interfaces
  - Supports Management Component Transport Protocol (MCTP) and Alert Standard Format (ASF) usage models
  - Interrupt generation capability for SMBAlert# signaling received from External SMBus device
  - Chapter 36, "System Management"

- LAN SMBus:
  - Pass-through and/or configuration traffic between an external Manageability Controller (MC) and Integrated Ethernet LAN ports
  - Operates as master or slave
  - Supports MCTP over SMBus
  - Supports ARP
  - Interface pins MUXed with the Network Controller Sideband Interface (NC-SI).
  - Chapter 13, “LAN Controllers”
- IE SMBuses:
  - BMC offload engine to access PECI services
  - Operates as master or slave
  - Chapter 37, “Innovation Engine”

§ §

# 28 SMBus Controller - Legacy

The SMBus controller described in this chapter is located in the Platform Control Cluster (PCC) of the SoC. In SoC diagrams, it is labeled Legacy SMBus.

The controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The SoC is also capable of operating in a mode that communicates with $I^2C$-compatible devices.

The SoC performs SMBus messages with Packet Error Checking (PEC) enabled or disabled. The actual PEC calculation and checking are performed in either the hardware or the software.

The SMBus Address Resolution Protocol (ARP) is supported by using the existing SMBus controller commands through the software. The controller can be the designated SMBus ARP Master.

This SMBus interface is shared with the Closed Loop Temperature Throttling (CLTT) circuitry to allow the CLTT to access the thermal information, if available, through the system memory DIMMs' Temperature Sensor On DIMM (TSOD) circuitry. See Section 35.5.1.5, "Legacy SMBus Used for Accessing Information for CLTT."

**Figure 28-1. What is Covered in This Chapter**

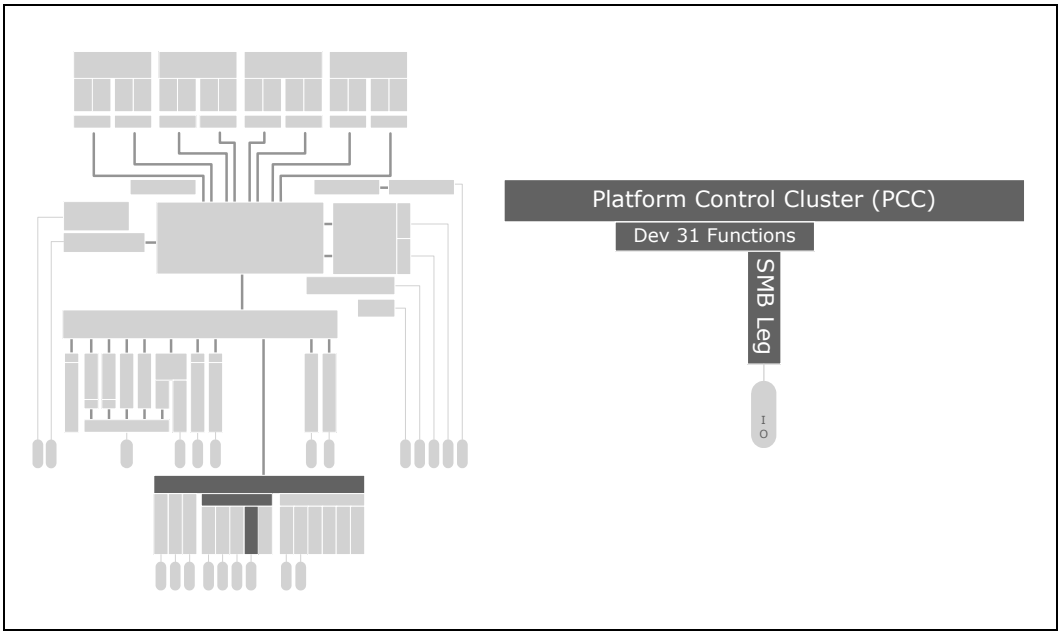



**Table 28-1. References**

| Document Title | Document ID / Location |
|---|---|
| *System Management Bus (SMBus) Specification, Version 3.0* | http://smbus.org/specs/SMBus_3_0_20141220.pdf |

## 28.1 Signal Descriptions

The signal descriptions are shown in Table 28-2, "Signal Names." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals, see Section 39.1, "Directory of Signal Names and Pin Names." The Direction/Type column of Table 28-2, "Signal Names" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 28-2. Signal Names**

| Signal Name | Direction | Shared | Description |
|---|---|---|---|
| SMB_LEG_CLK | I, O-OD | Yes | **SMBus Clock (SMBCLK)** |
| SMB_LEG_DATA | I, O-OD | Yes | **SMBus Data (SMBDAT)** |
| SMB_LEG_ALRT_N | I, O-OD | Yes | **SMBus Alert (SMBALERT#):** This signal wakes the system or generates a System Management Interrupt (SMI). |

The optional SMBus 2.0 signal, SMBSUS#, is not supported.

## 28.2 General Architecture

From a software driver perspective, the Legacy SMBus controller described in this chapter exists as a PCI device on Bus 0, Device 31 (decimal), Function 4 of configuration space. Its logic is controlled by the SoC RTC Clock. The controller's I/O registers are relocatable. The I/O base address registers are:

- SBA - SoC Host I/O Space
- SMBMBAR - SoC Host Memory Space

The registers and offsets are the same for SBA access and SMBAR access.

Key capabilities supported and not supported by the SoC Legacy SMBus controller:

- It is not supported as an SMBus Slave except when it is acting as the SMBus ARP Master. It functions as an SMBus Master as defined in the *System Management Bus (SMBus) Specification, Version 3.0*.
- Acting as the SMBus Host is not supported.

  As defined by the *System Management Bus (SMBus) Specification, Version 3.0*, an SMBus Host is a specialized SMBus Master that provides the main interface to the system's CPU. An SMBus Host must be an SMBus Master-Slave and must support the SMBus Host Notify protocol. There may be at most one SMBus Host in a system.
- Can be designated as the SMBus Address Resolution Protocol (ARP) Master.

  A system design usually defines the SMBus Host (defined above) to be the SMBus ARP Master but under some circumstances another SMBus Master, like the SoC Legacy SMBus controller, may be given the role by system software. There can only be one active SMBus ARP Master at any time.
- Functions as an SMBus version 2.0 Master or ARP Master device when the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register is 0.
- Functions as an $I^2C$ Master when the I2CEN bit is 1. Not all protocols are supported when I2CEN is 1. The protocol descriptions in this chapter describe what is supported. When the I2CEN bit is set, the SoC never uses its 32-byte buffer for any block commands.
  - The SoC supports an $I^2C$ Read command which is independent of the setting of the I2CEN bit.

## 28.3 Legacy SMBus Controller Operation

In order to initiate an SMBus operation as an SMBus Master, software sets up the registers of the SoC Legacy SMBus controller with a Slave Address (to which the operation is directed), a command to issue to the addressed Slave, and for write protocols, data to be sent to the Slave and an optional Packet Error Checking (PEC) bits. For read protocols, there is a designated SoC register for the data received from the Slave. The following registers are involved:

- Transmit Slave Address Register (TSA) - TSA bit field 7:1 is the seven-bit Slave address to be used of the targeted SMBus Slave. TSA bit 0 is 0 for write (to Slave) and 1 for read (from Slave). Transactions with TSA[7:0] values of 1010_xxx0 are blocked if the SPD Write Disable (SPDWD) bit = 1 in the Host Configuration (HCFG) register in Configuration Space. SPDWD prevents writing into the DIMM's SPD EEPROM.
- Host Command Register (HCMD) - Contains the eight-bit information to use as the transaction's SMBus command field. "Host" has nothing to do with the term "SMBus Host" mentioned earlier.
- Data 0 Register (HD0), Data 1 Register (HD1), and Host Block Data (HBD) - Used for the data transfer cycles of the particular protocol transaction.

Software then initiates the hardware operation by writing to the Start bit of the Host Control (HCTL) register. While the hardware executes the HCMD command, it sets the Host Busy (HBSY) bit of the Host Status Register Address (HSTS).

When completed, the hardware generates an interrupt or SMI to notify the software to access the HSTS register to view the status and error bits. If the command completed successfully, the Interrupt (INTR) bit of the HSTS register is set. Software must not alter the TSA, HCMD, HD0, HD1 and HBDregisters, until the INTR bit is set.

If the Slave device does not respond with an SMBus Acknowledge (ACK), or holds the SMBus Clock (SMB_LEG_CLK SoC signal pin) lower than the allowed timeout (25 ms which is 800 RTC clock cycles) the transaction times out. Here the hardware discards the cycle and the Device Error (DEVERR) bit is set. If the software sets the Kill (KILL) bit of the Host Control (HCTL) register is set by software while the command is running, the transaction stops and the Failed (FAIL) bit of the HSTS register is set.

The SoC Legacy SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be setup to generate an internal SMI# instead of an IRQ interrupt, by setting the SMB_SMI_EN (SSEN) bit of the HCFG register. More is said about this in Section 28.8, "Interrupts and SMI." Also see Section 8.1.1, "Interrupt Pin Route Overview" for PIRQB# routing. The SoC SMI is described in Section 8.3, "System Management Interrupt (SMI)."

### 28.3.1 Supported SMBus Protocols

The Legacy SMBus controller supports the SMBus protocols and $I^2C$ transactions shown in Table 28-3, "Supported SMBus ARP, SMBus, and $I^2C$ Protocols." For details of the bus cycles for each protocol, see the *System Management Bus (SMBus) Specification, Version 3.0*.

### Table 28-3. Supported SMBus ARP, SMBus, and I²C Protocols

| Protocol | Supported | |
|---|---|---|
| **SMBus ARP Commands** | **Sent as ARP Master** | **Received as ARP Slave** |
| Prepare to ARP | Yes, by Software Driver | Cannot be an SMBus slave |
| Reset Device<br>*General and directed* | | |
| Get UDID<br>*General and directed* | | |
| Assign Address | | |
| **SMBus ARP Commands** | **Sent as ARP Slave** | **Received as ARP Master** |
| Notify ARP Master | Cannot be an SMBus slave | Yes |
| **SMBus Commands** | **Sent as Master** | **Received as Slave** |
| Quick Command | Yes | Cannot be an SMBus slave |
| Send Byte | Yes | |
| Receive Byte | Yes | |
| Write Byte/Word | Yes | |
| Read Byte/Word | Yes | |
| Process Call | Yes | |
| Block Write | Yes | |
| Block Read | Yes | |
| Block Write-Block Read Process Call | Yes | |
| SMBus Host Notify | Yes | |
| **I²C Commands** | **Sent as Master** | **Received as Slave** |
| Master-TX writes to Slave-RX (MTx-to-SRx)<br>*No direction change* | Yes | Cannot be an I²C slave |
| Master-RX reads Slave-TX (STx-to-MRx) | Yes | |
| Combined format, write-then-read<br>*direction change after initial write* | No | |
| Combined format, read-then-write<br>*direction change after initial read* | No | |

## 28.3.2 SMBus CRC Generation and Checking

If the Automatically Append CRC (AAC) bit of the Auxiliary Control (AUXC) register is set, the SoC automatically calculates and drives the Cyclic Redundancy Check (CRC) at the end of the transmitted packet for write cycles and checks the CRC for read cycles. It does not transmit the contents of the Packet Error Check Data (PEC) PEC register for a CRC. The PEC Enable (PEC_EN) bit of the Host Control Register (HCTL) must not be set if the AAC bit is set, or else unspecified behavior results.

If the read cycle results in a CRC error, the Device Error (DERR) bit of the Host Status Register Address (HSTS) and the CRC Error (CRCE) bit of the Auxiliary Status (AUXS) register are set.

## 28.4 Protocols as an SMBus Master

Software sets up registers and instructs the SoC Legacy SMBus Controller to issue various protocols and commands. Programming details are presented here.

When the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register is 1, the SoC Legacy SMBus Controller is enabled to communicate with $I^2C$ devices. This is also called $I^2C$ mode and changes the formatting of some commands. When I2CEN is 0, behavior is for SMBus as defined in the *System Management Bus (SMBus) Specification, Version 3.0*. The command descriptions below mention the I2CEN bit and how it affects the transaction.

### 28.4.1 Quick Command

When the Legacy SMBus Controller is programmed to issue a Quick Command, the Transmit Slave Address (TSA) register is sent. The PEC byte is never appended to the quick protocol. The software forces the PEC Enable (PEC_EN) bit of the Host Control Register (HCTL) to 0 when performing the Quick Command. Software must force the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register to 0 when running this command. See Section 5.5.1 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

### 28.4.2 Send Byte and Receive Byte Commands

For the Send Byte command, the Transmit Slave Address (TSA) and Host Command (HCMD) registers are sent. For the Receive Byte command, the Transmit Slave Address (TSA) register is sent. The data received is stored in the Data 0 (HD0) register. Software must force the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register to bit to 0 when running this command.

The Receive Byte is similar to a Send Byte, the only difference is the direction of data transfer. See Sections 5.5.2 and 5.5.3 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

### 28.4.3 Write Byte and Write Word Commands

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a Write Byte/Word command, the Transmit Slave Address (TSA), Host Command (HCMD), and Data 0 (HD0) registers are sent. In addition, the Data 1 (HD1) register is sent on a Write Word command. Software must force the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register to 0 when running this command. See Section 5.5.4 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

### 28.4.4 Read Byte and Read Word Commands

Reading data is slightly more complicated than writing data. First the SoC must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device address. The slave then returns 1 or 2 bytes of data. Software must force the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register to 0 when running this command.

When programmed for the Read Byte/Word command, the Transmit Slave Address (TSA) and Host Command (HCMD) registers are sent. Data is received into the Data 0 (HD0) on the read byte, and the Data 0 (HD0) and Data 1 (HD1) registers on the read word. See Section 5.5.5 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

### 28.4.5 Process Call Command

The Process Call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the SoC transmits the Transmit Slave Address (TSA), Host Command (HCMD), Data 0 (HD0) and Data 1 (HD1) registers. Data received from the device is stored in the Data 0 (HD0) and Data 1 (HD1) registers.

The Process Call command produces undefined results when both the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register and the PEC Enable (PEC_EN) bit of the Host Control (HCTL) register are set. Software must force either the IC2EN bit to 0 or both the PCE_EN and the Automatically Append CRC (AAC) bits of the Auxiliary Control (AUXC) register to 0 when running this command. See Section 5.5.6 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

*Note:* For the Process Call command, the RW bit of the TSA register must to be programmed to 0 (indicating Write).

*Note:* If the I2CEN bit is set, the protocol sequence changes slightly. The command code (bits [18:11] in the bit sequence) is not sent, and as a result, the targeted Slave device does not Acknowledge (bit 19 in the sequence).

### 28.4.6 Block Read and Block Write Commands

The SoC contains a 32-byte buffer for read and write data which is enabled by setting the Enable 32-byte Buffer (E32B) bit of the Auxiliary Control (AUXC) register. If E32B is 0, single-byte buffering is used. This 32-byte buffer is filled with write data before transmission and filled with read data on reception. In the SoC, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

*Note:* If the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register is set, the SoC never uses the 32-byte buffer for any Block commands.

The byte count field is transmitted but ignored by the SoC as the software ends the transfer after all bytes it cares about have been sent or received.

For a Block Write, software must force either the IC2EN bit to 0 or both the PCE_EN and the Automatically Append CRC (AAC) bits of the Auxiliary Control (AUXC) register to 0 when running this command.

The Block Write begins with issuing a Slave Address and a Write indication. After the command code, the SoC issues a byte count describing how many more bytes follow in the message. If the Slave device has 20 bytes to send, the first byte is the number 20 (14h), followed by 20 bytes of data. The byte count is not 0. A Block Read or Write is allowed to transfer a maximum of 32-data bytes.

When programmed for a Block Write command, the Transmit Slave Address (TSA), Host Command (HCMD) and Data 0 (HD0) registers are sent. Data is then sent from the Host Block Data (HBD) register; the total data sent being the value stored in the Data 0 (HD0) register. On Block Read commands, the first byte received is stored in the Data 0 (HD0) register, and the remaining bytes are stored in the Host Block Data (HBD) register. See Section 5.5.7 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

*Note:* For a Block Write, if the I2CEN bit is set, the format of the command changes slightly. The SoC still sends the number of bytes (on writes) or receives the number of bytes (on reads) indicated in the Data 0 (HD0) register. However, it does not send the contents of the Data 0 (HD0) register as part of the message. Also, the Block Write protocol sequence changes slightly. The byte count (bits [27:20] in the bit sequence) is not sent, and as a result, the Slave device does not Acknowledge (bit 28 in the sequence).

## 28.4.7 Block Write-Block Read Process Call Command

The Block Write-Block Read Process Call command is a two-part message.

The call begins with a Slave Address and a Write indication. After the command code, the host issues a write byte count (M) that describes how many more bytes are written in the first part of the message. If a master has six bytes to send, the byte count field has the value six (0000_0110), followed by the six bytes of data. The write byte count (M) cannot be 0.

The second part of the message is a block of read data beginning with a repeated start condition followed by the slave address and a read bit. The next byte is the read byte count (N), which differs from the write byte count (M). The read byte count (N) cannot be 0.

The combined data payload must not exceed 32 bytes. The byte-length restrictions of this process call are summarized as follows:

- M ≥ 1 byte
- N ≥ 1 byte
- M + N ≤ 32 bytes

The read byte count does not include the Packet Error Checking (PEC) byte. The PEC is computed on the total message beginning with the first slave address and using the normal PEC computational rules. Intel recommends that a PEC byte be used with the Block Write-Block Read Process Call command. The software must read the Host Command (HCMD) register to reset the 32-byte buffer pointer before reading the Host Block Data (HBD) register.

*Note:* No STOP condition is before the repeated START condition, and that a NACK signifies the end of the read transfer.

*Note:* The Enable 32-byte Buffer (E32B) bit of the Auxiliary Control (AUXC) register must be set when using this protocol.

See Section 5.5.8 of the *System Management Bus (SMBus) Specification, Version 3.0* for the format of the protocol.

## 28.4.8 SMBus Host Notify Command

See Section 5.5.9 SMBus Host Notify Protocol of the *System Management Bus (SMBus) Specification, Version 3.0*.

The SoC Legacy SMBus Controller uses this protocol to communicate with the SMBus Host which has its Slave address defined as 0001_000.

## 28.5 Protocols and Commands as the ARP Master

The SoC Legacy SMBus controller and associated software driver can act as the SMBus ARP Master. Details of the role of the Address Resolution Protocol (ARP) Master are in Section 5.6.3 ARP Commands of the *System Management Bus (SMBus) Specification, Version 3.0*.

The ARP Master operation is supported via software that uses the existing Legacy SMBus controller commands in the Host Controller Command (HCMD) register in I/O Space and also in Memory Space. When acting as the ARP Master, bits [6:0] of the Receive Slave Address (RSA) register must be programmed to 0001_000.

The supported ARP commands are shown in Table 28-3, "Supported SMBus ARP, SMBus, and $I^2C$ Protocols." The recommended software-driver flow is shown in Figure 29-3, "ARP Master Behavior Flow Diagram." This diagram is taken from the *System Management Bus (SMBus) Specification, Version 3.0*.

### 28.5.1 ARP Enumeration and the Notify ARP Master Command

Any SMBus Master, including the Legacy SMBus Controller, can be delegated to perform ARP enumeration. Enumeration is the process that assigns Slave Addresses to the ARP-Capable Slave devices on the SMBus. Enumeration begins with a Master issuing the Prepare to ARP command as described in Section 5.6.3.2 Prepare to ARP of the *System Management Bus (SMBus) Specification, Version 3.0*. The controller performs a write to Slave Address 1100_001 which is the SMBus Device Default Address.

If it is the SMBus ARP Master (there can only be one on an SMBus interface), the SoC Legacy SMBus Controller is also responsible for responding to the Notify ARP Master directed to the Slave Address 0001_000 which is the Slave Address that can also be assigned to the SMBus Host.

If the Legacy SMBus Controller is not the ARP Master, because the SoC controller cannot act as an SMBus Slave, it has no Slave Address that can be assigned by the ARP Master and requires no ARP enumeration.

## 28.6 I²C Read Command

This command allows the SoC to perform Block Reads from certain I²C devices, such as serial EEPROMs. The SMBus Block Read supports the 7-bit addressing mode only. However, this does not allow access to devices using the I²C combined format that has data bytes after the address. Typically these data bytes correspond to an offset (address) within the serial memory chips.

*Note:* This command is supported independent of the setting of the I2C_EN (I2CEN) bit of the Host Configuration (HCFG) register. The I²C Read command with the PEC Enable (PEC_EN) bit set of the Host Control (HCTL) register produces undefined results. Undefined results also occurs when the The I²C Read command is started and the Append CRC (AAC) bits of the Auxiliary Control (AUXC) register is set.

For an I²C Read command, the RW bit of the TSA register RW must to be programmed to 1 (indicating Read). The format that is used for the command is shown in Table 28-4, "I²C Block Read."

**Table 28-4. I²C Block Read**

| Bit | Description |
|---|---|
| 1 | START indication driven by the SoC Legacy SMBus Controller |
| 8:2 | Slave Address – 7 bits |
| 9 | Write |
| 10 | Acknowledge (ACK) from the Slave device |
| 18:11 | Send Data 1 (HD1) register |
| 19 | ACK from the Slave device |
| 20 | Repeated START indication |
| 27:21 | Slave Address – 7 bits |
| 28 | Read |
| 29 | ACK from the Slave device |
| 37:30 | Data byte 1 from slave – 8 bits |
| 38 | ACK from the Slave device |
| 46:39 | Data byte 2 from the Slave device– 8 bits |
| 47 | ACK from the Slave device |
| – | Data bytes from the Slave device and associated ACKs[1] |
| – | Data byte N from the Slave device – 8 bits |
| – | NOT Acknowledge (NACK) |
| – | STOP indication driven by the SoC. |

1. The SoC continues reading data from the I²C Slave device until the NACK is received.

## 28.7 Bus Arbitration as an SMBus Master

See Section 4.3.2 Arbitration of the *System Management Bus (SMBus) Specification, Version 3.0*.

As an SMBus Master, if the SoC Legacy SMBus Controller starts to initiate a transaction on the bus and loses arbitration, the SoC sets the Bus Error (BERR) of the Host Status Register Address (HSTS) and, if enabled, generates an interrupt or SMI. The processor restarts the transaction.

When the controller wins arbitration, it drives the SMBCLK (the SoC SMB_LEG_CLK signal pin) during the transaction. The minimum frequency of the clock is 10 kHz and the maximum is 100 kHz. The SMBDAT (the SoC SMB_LEG_DATA signal pin) is driven or sensed by the SoC during the transaction defined by the protocol. The SoC also ensures minimum time between SMBus transactions it initiates.

### 28.7.1 SMBCLK Signal Stretching by an SMBus Slave

See Section 4.3.3 Clock low Extending of the *System Management Bus (SMBus) Specification, Version 3.0*.

The Slave device of the transaction may be slower than the Legacy SMBus Controller and may need additional time to respond to the controller. Slower Slave devices are allowed to stretch the low portion of the SMBCLK signal by driving it low (along with the Master) during the transfer of one message up to the maximum limits described in the AC specifications of the *System Management Bus (SMBus) Specification, Version 3.0*. The Slave device must not extend the SMBCLK too long and must maintain an SMBCLK period of no more than 100 µs (minimum SMBCLK frequency of 10 kHz). There are some exceptions to this and are mentioned in the SMBus specification.

### 28.7.2 SMBus Timeout

See Section 3.1.1.2 Timeouts of the *System Management Bus (SMBus) Specification, Version 3.0*.

If an error is in the transaction, such that an SMBus Slave device does not signal an Acknowledge (ACK) to the Legacy SMBus Controller, or holds the SMBCLK low (stretching) longer than the allowed timeout duration, the transaction times out. The SoC discards the cycle and sets the Device Error (DERR) bit of the Host Status Register Address (HSTS). The minimum timeout duration is 25 ms (800 RTC clock periods). The time-out counter inside the SoC starts after the first bit of data is transferred. The 25-ms time-out counter does not count when the Byte Done Status bit (BYTE_DONE_STS) of the Host Status Register Address (HSTS) is set.

## 28.8 Interrupts and SMI

The SMBus controller uses PIRQB# as its interrupt pin. However, the system is alternatively set up to generate a System Management Interrupt (SMI) instead of an interrupt, by setting the SMB_SMI_EN (SSEN) bit of the Host Configuration (HCFG) Register.

The following tables specify how the various enable bits in the SMBus function control the generation of the interrupt, host SMI, and wake internal signals:

- Table 28-5, “Enable for SMBALERT# (SMB_LEG_ALRT_N)”
- Table 28-6, “Enables for Legacy SMBus Controller Events”
- Table 28-7, “Enables for a Received Notify ARP Master Command”

The rows in the tables are additive, which means that if more than one row is true for a particular scenario then the results for all of the activated rows occur.

### Table 28-5. Enable for SMBALERT# (SMB_LEG_ALRT_N)

| Event | INTREN (INTREN) bit of the Host Control (HCTL) Register | SMB_SMI_EN (SSEN) bit of the Host Configuration (HCFG) Register | SMBALERT_DIS (SMB_D) bit of the Slave Command (SCMD) Register | Result |
|---|---|---|---|---|
| SMBALERT# (SMB_LEG_ALRT_N) asserted low<br>*Always reported in SMBALERT bit of the Host Status Register Address (HSTS)* | X | X | X | Wake Event generated to PMC via the SMB_WAK_STS bit of the General Purpose Event 0 Status [127:96] (GPE0_STS_127_96) register of the SoC PMC.<br>*Depends on Intel® Management Engine settings* |
| | X | 1 | 0 | Besides the Wake Event, an SMI is generated by the Legacy SMBus Controller via the SMBALERT_STS of the Host Status Register Address (HSTS) register.<br>Also, when in the S0 state, this also sets SMBUS_SMI_STS bit of the SMI Status (SMI_STS) register of the SoC PMC. |
| | 1 | 0 | 0 | IRQ Interrupt generated |

### Table 28-6. Enables for Legacy SMBus Controller Events

| Event | INTREN (INTREN) bit of the Host Control (HCTL) Register | SMB_SMI_EN (SSEN) bit of the Host Configuration (HCFG) Register | Result |
|---|---|---|---|
| Any combination of FAIL, BERR, DERR, and INTR bits of the Host Status Register Address (HSTS) provided they are individually enabled to generate and interrupt. | 0 | X | None |
| | 1 | 0 | IRQ interrupt generated |
| | 1 | 1 | SMI generated to the PMC when in S0 state via the LEGACY_USB2_STS bit of the SMI Status (SMI_STS) register of the SoC PMC. |

### Table 28-7. Enables for a Received Notify ARP Master Command

| HOST_NOTIFY_INTREN (HNI) bit of the Slave Command (SCMD) Register | SMB_SMI_EN (SSEN) bit of the Host Configuration (HCFG) Register | HOST_NOTIFY_WKEN (HNW) bit of the Slave Command (SCMD) Register | Result |
|---|---|---|---|
| 0 | X | 0 | None |
| X | X | 1 | Wake Event generated to PMC via the SMB_WAK_STS bit of the General Purpose Event 0 Status [127:96] (GPE0_STS_127_96) register.<br>*Depends on Intel® Management Engine settings* |
| 1 | 0 | X | IRQ interrupt generated |
| 1 | 1 | X | SMI generated to the PMC when in S0 state via the SMBUS_SMI_STS bit of the SMI Status (SMI_STS) register of the SoC PMC. |

## 28.9 Optional SMBALERT# Signal

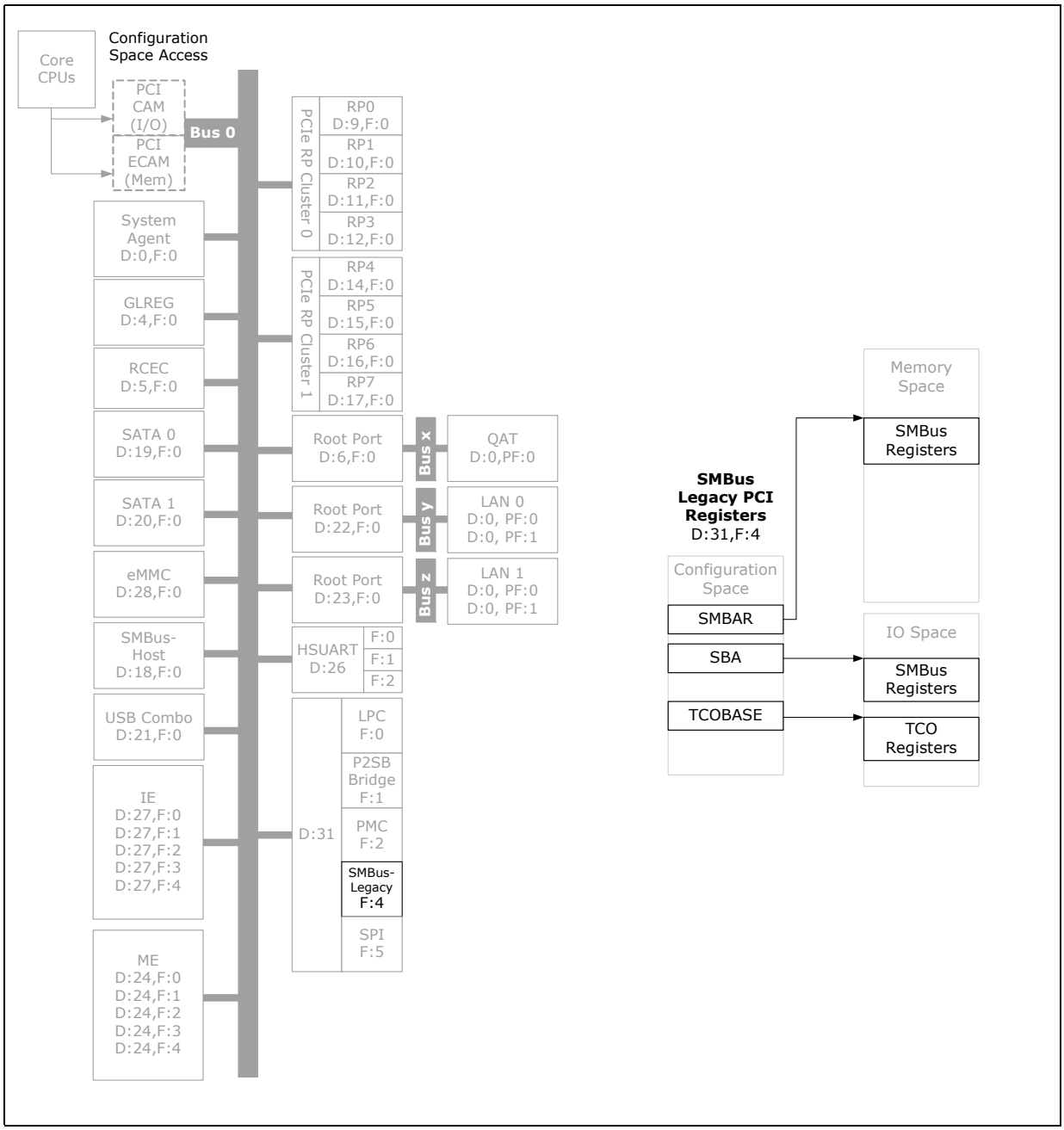
The optional SMBALERT# signal is multiplexed with Customer GPIO[112]. If not blocked by the SMBALERT_DIS (SMB_D) bit of the Slave Command (SCMD) register, the SoC generates an interrupt or an SMI when SMBALERT# is asserted by an SMBus Slave device. This signal is a wake event from the system S4 or S5 (soft off) power state. See Table 28-5, "Enable for SMBALERT# (SMB_LEG_ALRT_N)." The SMBUSALERT# signal is described in the Appendix A – Optional SMBus Signals section of the *System Management Bus (SMBus) Specification, Version 3.0*.

The interrupt is meant to alert software to setup the Legacy SMBus controller to issue a message over the SMBus with the slave address set to the Alert Response Address (0001_000). This message accesses all SMBUSALERT# devices on the bus. There could be more than one Slave device asserting the SMB_LEG_ALRT_N signal. The controller observes the returned Slave-device address(es) and performs device-address arbitration as defined in the *System Management Bus (SMBus) Specification, Version 3.0* to determine which SMBus Slave device shall be attributed to the alert.

## 28.10 Register Map

Figure 28-2, "Register Map" shows the associated registers from a system software viewpoint. The TCO Registers are used by the SoC Power Management Controller. See Chapter 16, "Power Management Controller (PMC)."

**Figure 28-2. Register Map**



## 28.10.1 Registers in the Configuration Space

### Table 28-8. Registers in the Configuration Space

| Offset | Name | Description |
|---|---|---|
| 0h | VID | Vendor ID |
| 2h | DID | Device ID |
| 4h | CMD | Command |
| 6h | DS | Device Status |
| 8h | RID | Revision ID |
| 9h | PI | Programming Interface |
| Ah | SCC | Sub Class Code |
| Bh | BCC | Base Class Code |
| 10h | SMBMBAR_31_0 | SMBus Memory Base Address_31_0 |
| 14h | SMBMBAR_63_32 | SMBus Memory Base Address_63_32 |
| 20h | SBA | SMB Base Address |
| 2Ch | SVID | Subsystem Vendor ID |
| 2Eh | SID | Subsystem ID |
| 3Ch | INTLN | Interrupt Line |
| 3Dh | INTPN | Interrupt Pin |
| 40h | HCFG | Host Configuration |
| 50h | TCOBASE | TCO Base Address |
| 54h | TCOCTL | TCO Control |
| F8h | MANID | Manufacturer's ID |

## 28.10.2 Registers in the Memory Space

### Table 28-9. Registers in the Memory Space

| Offset | Name | Description |
|---|---|---|
| 0h | HSTS | Host Status Register Address |
| 2h | HCTL | Host Control Register |
| 3h | HCMD | Host Command Register |
| 4h | TSA | Transmit Slave Address Register |
| 5h | HD0 | Data 0 Register |
| 6h | HD1 | Data 1 Register |
| 7h | HBD | Host Block Data |
| 8h | PEC | Packet Error Check Data Register |
| 9h | RSA | Receive Slave Address Register |
| Ah | SD | Slave Data Register |
| Ch | AUXS | Auxiliary Status |
| Dh | AUXC | Auxiliary Control |
| Eh | SMLC | SMLINK_PIN_CTL Register |
| Fh | SMBC | SMBUS_PIN_CTL Register |
| 10h | SSTS | Slave Status Register |
| 11h | SCMD | Slave Command Register |
| 14h | NDA | Notify Device Address Register |
| 16h | NDLB | Notify Data Low Byte Register |
| 17h | NDHB | Notify Data High Byte Register |

## 28.10.3 Registers in the I/O Space

### Table 28-10. Registers in the I/O Space

| Offset | Name | Description |
|---|---|---|
| 0h | HSTS | Host Status Register Address |
| 2h | HCTL | Host Control Register |
| 3h | HCMD | Host Command Register |
| 4h | TSA | Transmit Slave Address Register |
| 5h | HD0 | Data 0 Register |
| 6h | HD1 | Data 1 Register |
| 7h | HBD | Host Block Data |
| 8h | PEC | Packet Error Check Data Register |
| 9h | RSA | Receive Slave Address Register |
| Ah | SD | Slave Data Register |
| Ch | AUXS | Auxiliary Status |
| Dh | AUXC | Auxiliary Control |
| Eh | SMLC | SMLINK_PIN_CTL Register |
| Fh | SMBC | SMBUS_PIN_CTL Register |
| 10h | SSTS | Slave Status Register |
| 11h | SCMD | Slave Command Register |
| 14h | NDA | Notify Device Address Register |
| 16h | NDLB | Notify Data Low Byte Register |
| 17h | NDHB | Notify Data High Byte Register |

§ §

# 29 SMBus Controller - Host

The SMBus Message Transport (SMT) controller provides a mechanism whereby the SoC sends and receives out-of-band manageability messages over the SMBus to managed devices or to management controllers. This allows it to participate in manageability services with Intel and third-party devices like embedded controllers, sensors, and other devices through the SMBus interface.

**Figure 29-1. What is Covered in This Chapter**

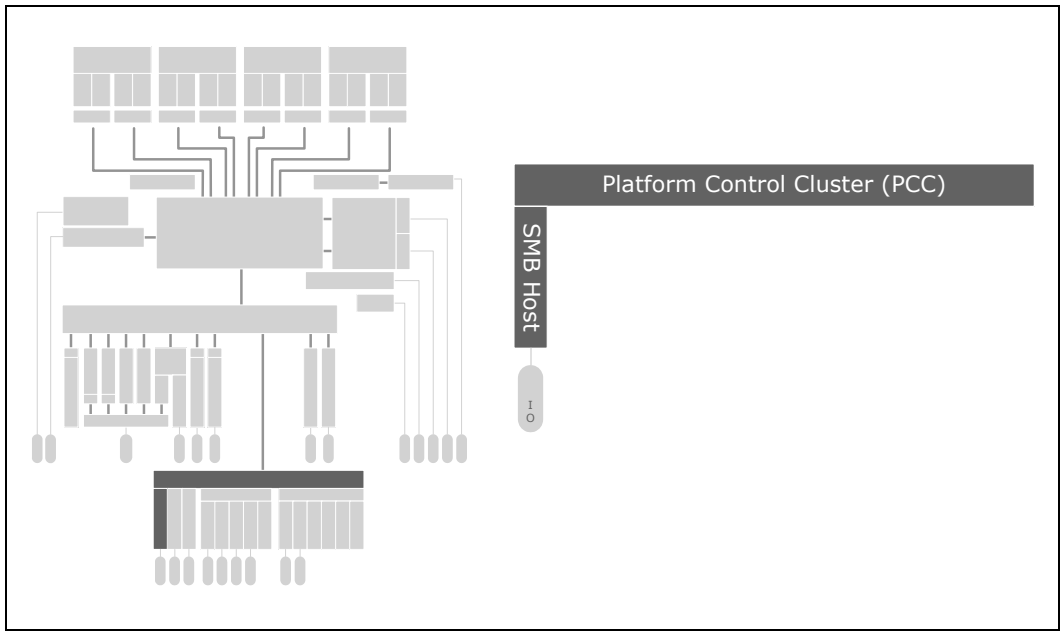



**Table 29-1. References**

| Document Title | Document ID / Location |
|---|---|
| *System Management Bus (SMBus) Specification, Version 3.0* | http://smbus.org/specs/SMBus_3_0_20141220.pdf |
| *Alert Standard Format (ASF) Specification, Version 2.0* | https://www.dmtf.org/sites/default/files/standards/documents/DSP0136.pdf |
| *Management Component Transport Protocol (MCTP) Base Specification, Version 1.1.0* | https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.1.0.pdf |

## 29.1 Signal Descriptions

The signal descriptions are shown in Table 29-2. For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a customer GPIO signals, see Section 39.1, "Directory of Signal Names and Pin Names" on page 993. The Direction/Type column of Table 29-2 is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 29-2. Signal Names**

| Signal Name | Direction Type | Shared | Description |
|---|---|---|---|
| SMB_HOST_CLK | I, O-OD | Yes | SMBus Clock (SMBCLK) |
| SMB_HOST_DATA | I, O-OD | Yes | SMBus Data (SMBDAT) |

The optional SMBus 2.0 signals, SMBALERT# and SMBSUS#, are not supported on this controller interface.

## 29.2 Features

- Operates as an SMBus master or slave.
- Supports ARP in the master or slave mode.
- Packet Error Code (PEC) is enabled for SMBus transactions.
- Compatible with certain $I^2C$ master (all commands except read-then-write) and slave (only MTx-to-SRx) modes.
- Status and errors are communicated by polling or interrupts.
- Supports INTx or Message Signaled Interrupt (MSI).

## 29.3 Architectural Overview

The SMBus Message Transport (SMT) controller has three regions in the configuration space. These are discovered by software in the configuration space as bus 0, device 18 (decimal), function (0). The address offsets and capability IDs of the contents of these regions are:

1. PCI Standard Header
   - Type 0
2. PCI Capabilities List
   - 40h: PCI Express* - Capability ID = 10h
   - 80h: PCI Power Management - Capability ID = 01h
   - 8Ch: Message Signaled Interrupts (MSI) - Capability ID = 05h
   - Various implementation-specific and Intel-reserved registers
3. PCI Express Extended Capabilities List
   - 100h: Advanced Error Reporting (AER) - Extended Capability ID = 0001h

The SMT controller operation is DMA-based in which descriptors and data are exchanged between the SoC hardware and the firmware through system memory. This DMA mode is available for the SMT acting both as the master and target.

As transport layer functionality, SMT transfers messages between the devices on the SMBus segment (to which it is physically connected) and the SoC firmware. The SMT hardware is physically located within the SoC and resides within its PCIe* space. In the PCIe hierarchy, SMT is bus 0, device 18 (decimal), function 0.

Table 29-3 summarizes which SMBus ARP, SMBus, and I2C protocols are supported by the SMT controller.

**Table 29-3. List of Supported SMBus ARP, SMBus, and I2C Protocols**

| Protocol | Supported | |
|---|---|---|
| **SMBus ARP Commands** | **Sent as ARP Master** | **Received as ARP Slave** |
| Prepare to ARP | Yes | |
| Reset Device<br>*General and directed* | | |
| Get UDID<br>*General and directed* | | |
| Assign Address | | |
| **SMBus ARP Commands** | **Sent as ARP Slave** | **Received as ARP Master** |
| Notify ARP Master | Yes | |
| **SMBus Commands** | **Sent as Master** | **Received as Slave** |
| Quick Command | Yes | Yes |
| Send Byte | Yes | Yes |
| Receive Byte | Yes | No |
| Write Byte/Word | Yes | Yes |
| Read Byte/Word | Yes | No |
| Process Call | Yes | No |
| Block Write | Yes | Yes |
| Block Read | Yes | Yes |
| Block Write-Block Read Process Call | Yes | No |
| SMBus Host Notify | Yes | Yes |
| **I2C Commands** | **Sent as Master** | **Received as Slave** |
| Master-TX writes slave-RX (MTx-to-SRx)<br>*No direction change* | Yes | Yes |
| Master-RX reads slave-TX (STx-to-MRx) | Yes | No |
| Combined format, write-then-read<br>*direction change after initial write* | Yes | No |
| Combined format, read-then-write<br>*direction change after initial read* | No | No |

## 29.4 Controller Characteristics and Operation

### 29.4.1 SMBus Behavior on PCIe Reset

When power is applied to an SMBus device, it performs default initialization of the internal state as specified in the *System Management Bus (SMBus) Specification, Version 3.0*. The SMBus device interface logic is not affected by PERST#. This normally allows the SMBus to support communications when the PCIe interface cannot.

### 29.4.2 Addressing and Configuration

An Address Resolution Protocol (ARP) is defined in the *System Management Bus (SMBus) Specification, Version 3.0* as assigning slave addresses to the SMBus devices. It is required that systems that connect the SMBus to PCIe slots implement the ARP for assignment of SMBus slave addresses to the SMBus interface devices on the PCIe add-in cards. The system must execute the ARP on a logical SMBus whenever any PCIe device in an individual slot associated with the logical SMBus exits the $D3_{COLD}$ state. Before executing the ARP, the system must ensure that all ARP-capable SMBus interface devices are returned to their default address state.

### 29.4.2.1 ARP Nomenclature

The following are some definitions pertaining to SMBus ARP.

**Table 29-4. ARP Nomenclature**

| Term | Definition |
|---|---|
| Address Resolution Protocol (ARP) | A protocol by which SMBus devices with assignable addresses on the bus are enumerated and assigned non-conflicting slave addresses. |
| Address Resolved (AR) Flag | A flag bit or state internal to the SoC that indicates whether or not the device slave address has been resolved by the ARP master. |
| Address Valid (AV) Flag | A flag bit or state internal to the SoC that indicates whether or not the device slave address is valid. This bit must be non-volatile for devices that support the persistent slave address. |
| ARP Master | The SMBus master executes the ARP and assigns addresses to the ARP-capable slave devices. The SMBus host usually is the ARP master, but under some circumstances another SMBus master assumes the role. Only one active ARP master exists at any time. |
| Persistent Slave Address (PSA) | An assigned slave address that is retained through the loss of device power. |
| SMBus Device Default Address | The address of all the ARP-capable slave devices must respond to. After a slave address has been assigned, a device must still respond to commands at the SMBus device default address for ARP management. This address is fixed at 1100 001. |
| SMBus Host | A specialized SMBus master that provides the main interface to the system CPU. It must be a master-slave and must support the SMBus host notify protocol. At most, one host is in a system. |
| Unique Device Identifier (UDID) | A 128-bit value that a device uses during the ARP process to uniquely identify itself. |

**Table 29-5. Device Decodes of AV and AR Flags**

| Address Valid (AV) | Address Resolved (AR) | Meaning |
|---|---|---|
| Cleared | Cleared | The device does not have a valid slave address and participates in the ARP process. This is the POR state for a device that does not support the Persistent Slave Address (PSA) or if it does it has not previously been assigned a slave address. |
| Cleared | Set | ILLEGAL STATE |
| Set | Cleared | The device has a valid slave address but still must participate in the ARP process. |
| Set | Set | The device has a valid slave address that has been resolved by the ARP master. The device does not respond to the Get UDID (general) command. However, it subsequently receives an Assign Address command and changed its slave address accordingly. |

### 29.4.2.2 Unique Device Identifier (UDID) Format

The UDID is a 128-bit value that a device uses during the ARP process to uniquely identify itself.

**Table 29-6. UDID Format**

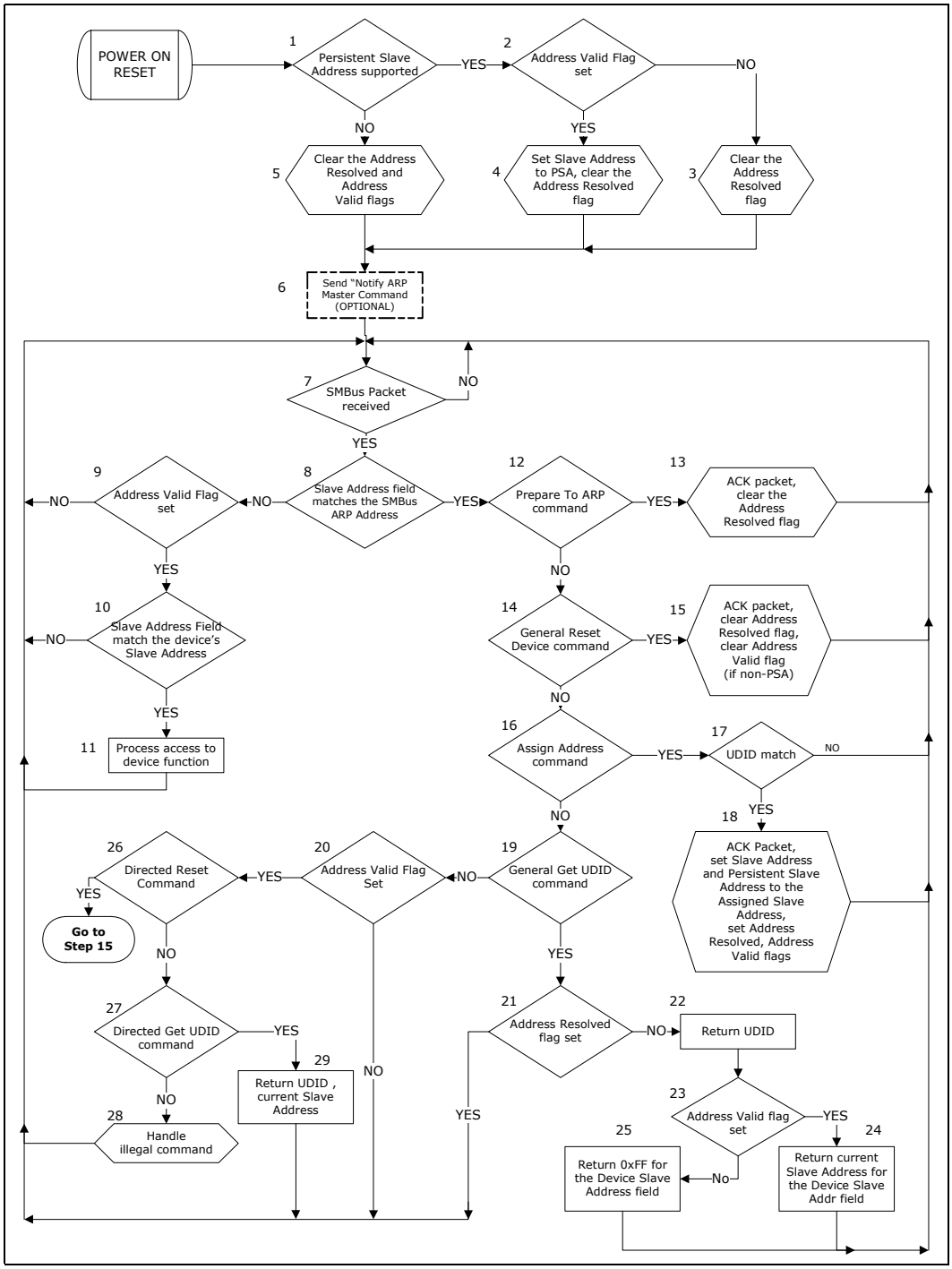
| Size | Field | Comments |
|---|---|---|
| 8 bits Most Significant Bit (MSB) | Device Capabilities | Includes the selection of a fixed or dynamic ARP address |
| 8 bits | Version/Revision | |
| 16 bits | Vendor ID | |
| 16 bits | Device ID | |
| 16 bits | Interface | |
| 16 bits | Subsystem Vendor ID | |
| 16 bits | Subsystem Device ID | |
| 32 bits Least Significant Bit (LSB) | Vendor Specific ID | Includes bits to uniquely identify each device UDIDs |

The vendor-specific field provides a unique ID for functionally equivalent devices. This field is for devices that otherwise return identical UDIDs for the purpose of the address assignment. A unique ID in this field is required since this device supports an assigned slave address. For a pre-assigned unique ID, at least 24 bits must be unique; however, the full 32 bits is recommended. Uniqueness is important to guarantee that two like devices are identified discretely.

The Vendor Specific ID (VSID) fields of the UDID0 Data Register (UDID0) and the UDID1 Data Register (UDID1) are located in the SMT controller MMIO.

### 29.4.2.3 ARP Slave Behavior

**Figure 29-2. ARP-Capable (Slave) Device Behavior Flow Diagram**




With reference to Table 29-7 the ARP slave operates as follows (steps which are shaded are the responsibility of the firmware alone or of the firmware to trigger the hardware).

## Table 29-7. ARP Slave Operations (Sheet 1 of 3)

| Step | | Description |
|---|---|---|
| 0. | Perform ARP slave initialization. | • [FW[1]] See Section 29.4.2.5, "ARP Initialization Flow" on page 746 |
| 1. | After exiting the power-on-reset state, a device that supports PSA goes to step 2 to verify if its slave address is valid. If the device does not support PSA, proceed to step 5. | • [FW] SoC does not support PSA; jump past PSA-related steps to step 5. |
| 2. | A device supporting PSA must check its address valid flag which is non-volatile. If that flag is set then it has previously received an assigned slave address; proceed to step 4. If the address valid flag is cleared then proceed to step 3. | • N/A |
| 3. | Although the device supports PSA the value is currently invalid. The device must clear the address resolved flag to indicate that it has not had its slave address assigned. Proceed to step 6. | • N/A |
| 4. | The device has a valid PSA so it assumes that slave address for now. However, this address has not been resolved by the ARP master so the device must clear its address resolved flag. Proceed to step 6. | • N/A |
| 5. | The device does not support PSA so it must clear its address valid and address resolved flags. Proceed to step 6. | • [FW] De-assert SMTARPCTRL.AV*n*.<br>• [FW] De-assert SMTARPCTRL.AR*n*. |
| 6. | If supported, the device masters the SMBus and sends the Notify ARP Master command. This informs the ARP master that a new device is present. Proceed to step 7. | • [FW] Determine if supported; check SMTARPCTRL.NOTIFYENB.<br>• [FW and HW[2]] Initiate master transaction (see Section 29.4.6.4, "Master Transactions Flow" on page 758). |
| 7. | The device waits for an SMBus packet. | • [FW and HW] Wait |
| 8. | Upon receipt of an SMBus packet the device must first check the received slave address against the SMBus device default address. If a match is present, then proceed to step 12, otherwise proceed to step 9. | • [HW] Match the default address.<br>• [HW] ACK the transaction if received slave address matches device default address. |
| 9. | The received address is not the SMBus device default address so the packet is potentially addressed to one of the device core functions. The device must check its Address Valid bits to determine whether or not to respond. If any of the Address Valid bits are set then proceed to step 10, otherwise return to step 7 and wait for another SMBus packet. | • [HW] Check if any SMTARPCTRL.AV*n* are asserted. |
| 10. | Since the device has a valid slave address it must compare the received slave address to its internal slave address. If a match is present, then proceed to step 11, otherwise return to step 7 and wait for another SMBus packet. | • [HW] Match any of the internal slave addresses.<br>• [HW] ACK the transaction if received slave address matches **any** target address **and** that target Address Valid bit (SMTARPCTRL.AV*n*) is asserted. |
| 11. | The device has received a packet addressed to a core function so it acknowledges the packet and processes it accordingly. Proceed to step 7 and wait for another SMBus packet. | • [FW and HW] Initiate target transaction (see Section 29.4.7.4, "Target Memory Buffer Hardware-Firmware Flow" on page 772). |
| 12. | The device detected a packet addressed to the SMBus Device default address. Check the command field to determine if this is the Prepare To ARP command. If so, then proceed to step 13, otherwise proceed to step 14. | • [FW] Inspects command field and determines how to proceed. |
| 13. | Upon receipt of the Prepare To ARP command the device must acknowledge the packet and make sure its address resolved flag is clear to participate in the ARP process. Proceed to step 7 and wait for another SMBus packet. | • [FW] De-assert SMTARPCTRL.AR*n*. |
| 14. | The device checks the command field to verify if the Reset Device command was issued. If so, then proceed to step 15, otherwise proceed to step 16. | • [FW] Inspects command field and determines how to proceed. |

## Table 29-7. ARP Slave Operations (Sheet 2 of 3)

| Step | | Description |
|---|---|---|
| 15. | Upon receipt of the Reset Device command the device must acknowledge the packet and make sure its address valid (if non-PSA) and address resolved flags are cleared. This allows the ARP master to re-assign all device addresses without cycling power. Proceed to step 7 and wait for another SMBus packet. | • [FW] De-assert SMTARPCTRL.AV*n*.<br>• [FW] De-assert SMTARPCTRL.AR*n*. |
| 16. | The device checks the command field to verify if the Assign Address command was issued. If so, then proceed to step 17, otherwise proceed to step 19. | • [FW] Inspects the command field and determines how to proceed. |
| 17. | Upon receipt of the Assign Address command the device must compare its UDID to the one it is receiving. If any byte does not match then it must not acknowledge that byte or subsequent ones. If all bytes in the UDID compare then proceed to step 18, otherwise return to step 7 and wait for another SMBus packet. | • [HW] Check the command field **only if** received slave address matches device default address.<br>• [HW] If the Assign Address command, then verify each byte of received UDID against all SMT UDIDs.<br>• [HW] NACK the transaction if received UDID fails to match any SMT UDID.<br>***Note:*** All ARP protocols are directed only to the device default address.<br>***Note:*** SMT must respond to this command even if its SMTARPCTRL.AR flag is set (i.e., ARP master overwrites a valid and resolved target address).<br>***Note:*** This also implies the hardware cannot detect an ARP-pending flag to selectively monitor the command field for Assign Address command.<br>***Note:*** If UDID is NACKed, the hardware is not required to notify the firmware. |
| 18. | Since the UDID matched, the device must assume the received slave address and update its PSA, if supported. The device must set its address valid and address resolved flags; this indicates it no longer responds to the Get UDID command unless it receives the Prepare To ARP or Reset Device commands or is power cycled. Proceed to step 7 and wait for another SMBus packet. | • [FW] Assign the received slave address to the matched UDID.<br>• [FW] Assert SMTARPCTRL.AV*n*.<br>• [FW] Assert SMTARPCTRL.AR*n*. |
| 19. | The device checks the command field to verify if the Get UDID command was issued. If so, then proceed to step 21, otherwise proceed to step 20. | • [HW] Inspects the command field and determines how to proceed. |
| 20. | The device is receiving a directed command. Directed commands must be acknowledged only by slaves with a valid address. If the address is not valid then ignore the packet and return to step 7 and wait for another SMBus packet. If the address is valid then proceed to step 26. | • [HW] Inspects the address valid flag. |
| 21. | Upon receipt of the Get UDID command, the device must check its address resolved flag to determine whether or not it participates in the ARP process. If set then its address has already been resolved by the ARP master so the device proceeds to step 7 to wait for another SMBus packet. If the AR flag is cleared then the device proceeds to step 22. | • [HW] Inspects the address resolved flag and determines how to proceed. |
| 22. | The device returns its UDID and monitors the SMBus data line for collisions. If a collision is detected at any time the device must stop transmitting and proceed to step 7 and wait for another SMBus packet. If no collisions were detected then proceed to step 23. | • [HW] UDID is returned as read data to master. |
| 23. | The device must now check its address valid flag to determine what value to return for the device slave address field. If the AV flag is set then proceed to step 24, otherwise proceed to step 25. | • [HW] Inspects the address valid flag and determines how to proceed. |
| 24. | The current slave address is valid so the device returns this for the device slave address field (with bit 0 set) and monitors the SMBus data line for collisions (i.e., another device driving a 0 when this device is driving a 1). Proceed to step 7 and wait for another SMBus packet. | • [HW] UDID and current slave address are returned as read data to master (as per generic block-read payload). |
| 25. | The current slave address is invalid so the device returns a value of 0xFF and monitors the SMBus data line for collisions. If the ARP master receives the 0xFF value it knows that the device requires address assignment. Proceed to step 7 and wait for another SMBus packet. | • [HW] UDID and 0xFF address are returned as read data to master (as per generic block-read payload). |

## Table 29-7. ARP Slave Operations (Sheet 3 of 3)

| Step | | Description |
|---|---|---|
| 26. | Is this a directed Reset Device command? If so then proceed to step 15. Otherwise proceed to step 27. | • [FW] Inspects the command field and determines how to proceed. |
| 27. | Is this the Get UDID (directed) command? If so then proceed to step 29. Return the UDID information. If not, then proceed to step 28. | • [HW] Inspects the command field and determines how to proceed. |
| 28. | The device has not received a valid command so it must handle the illegal command in accordance with SMBus rules for error handling. Proceed to step 7 and wait for another SMBus packet. | • [FW] Perform error handling. |
| 29. | Return the UDID information and current slave address, then proceed to step 7 and wait for another SMBus packet. | • [HW] UDID and current slave address are returned as read data to master (as per generic block-read payload). |

1. FW means Firmware.
2. HW means Hardware.

Table 29-8 identifies the hardware pattern-matching which is required to decode ARP and ordinary SMBus protocols. The UDID matching applies only to the assign address protocol.

Since ARP protocols are typically writes to the ARP slave by the ARP host, the R/W# bit (LSB) of the first byte is usually 0; however, the general and directed forms of Get UDID require the ARP slave to return its UDID to the ARP host, such that the LSB of the first byte after the repeated start is 1. During either Get UDID protocol, the hardware matching must detect the repeated start and subsequently match a slave address of C3h.

Likewise, most ordinary SMBus protocols begin with a write to the slave/target by the master. However, since certain protocols begin with a read to the slave, the hardware matching treats the LSB of the first byte as a (don't care). For those protocols which require the slave to return data to the master, the LSB of the first byte after the repeated start is 1. Therefore, the hardware matching must detect the repeated start and subsequently match a slave address with an appended 1 LSB.

## Table 29-8. Hardware Decoding of ARP, SMBus, and I²C Target Transactions

| Byte 1<br>Slave Address[1] | Byte 2<br>ARP Command | Bytes 4-19<br>UDID | Hardware-Decoded<br>Protocol |
|---|---|---|---|
| ARP protocols applicable to all SMT UDIDs | | | |
| C2h | 01h | N/A | Prepare to ARP |
| C2h | 02h | N/A | Reset Device (general) |
| C2h | 03h | N/A | Get UDID (general) |
| C3h[2] | N/A | N/A | Get UDID (general) —after repeated start |
| ARP protocols specific to only matched UDID (128 bits) | | | |
| C2h | 04h | Matches an SMT UDID | Assign Address |
| ARP protocols specific to only matched slave address (7 bits) | | | |
| C2h | {target address, 1} | N/A | Get UDID (directed) |
| C3h[2] | N/A | N/A | Get UDID (directed) —after repeated start |
| C2h | {target address, 0} | N/A | Reset Device (directed) |
| 10h | C2h | N/A | Notify ARP master |
| 10h | != C2h[3] | N/A | SMBus Host Notify |
| Ordinary SMBus (and I²C) protocol specific to only matched slave address (7 bits) | | | |
| {slave address, x} | N/A | N/A | Ordinary SMBus transaction.<br>LSB = x to be protocol agnostic<br>• Quick Command<br>• Send Byte<br>• Write Byte/Word<br>• Block Write<br>• Block Read |
| {slave address, **1**}[2] | N/A | N/A | Ordinary SMBus transaction.<br>LSB = 1 —after repeated start<br>• Block Read |

1. Hardware checking of LSB during start address cycle is controlled by SUSCHKB.IRWST. (Applies to SMBus target addresses, C2h, and 10h.) See also Section 29.4.7.5, "Target Flow" on page 775 and Table 29-20, "Target Transaction Behavior Due to SUSCHKB.IRWST" on page 776 for more implications to the firmware and the hardware regarding IRWST.
2. Hardware checking of LSB during repeated-start address cycle is controlled by SUSCHKB.IRWRST. (Applies to SMBus target addresses and C2h.)
3. For a transaction directed to 10h if the command byte is not C2h then the transaction is SMBus host notify and the command code format is {initiator slave address, 0}.

Certain SMBus protocols are disallowed in the target mode since they violate the hardware-firmware descriptor mechanism. First, the descriptor is passed to the firmware only after the transaction stop; this architecture does not permit the firmware to return a payload within the same transaction. Second, waiting for a descriptor from the firmware requires the hardware to perform excessive SMBus clock-stretching.

Table 29-9 captures the responses of the hardware/firmware under the various combinations of ARP protocol, address and UDID fields, and ARP status flags. The states mentioned refer to Table 29-9.

### Table 29-9. Hardware/Firmware Response to SMBus and ARP Protocols

| Received ARP Protocol | AV Flag | AR Flag | ARP States (1-8,...) | SoC Hardware and Firmware Response |
|---|---|---|---|---|
| Ordinary SMBus protocol | | | | |
| Ordinary SMBus protocol | T | N/A | 9, 10, 11 | Normal hardware-firmware flow |
| SMBus block-read | T | N/A | 9, 10, 11 | The hardware returns the specified number of bytes from the hardware buffers. |
| ARP protocol | | | | |
| Prepare to ARP | N/A | N/A | 12, 13 | Normal hardware-firmware flow |
| Reset Device (general) | N/A | N/A | 12, 14, 15 | Normal hardware-firmware flow |
| Assign Address | N/A | N/A | 12, 14, 16, 17, 18 | Normal hardware-firmware flow |
| Get UDID (general) | N/A | T | 12, 14, 16, 19, 21 | The hardware does NACK the ARP command byte to indicate it has a valid assigned slave address (ARP is complete). |
| Get UDID (general) | T | F | 12, 14, 16, 19, 21, 22, 23, 24 | The hardware returns its UDID and corresponding slave address. |
| Get UDID (general) | F | F | 12, 14, 16, 19, 21, 22, 23, 25 | The hardware returns its UDID and 0xFF for its address. |
| Reset Device (directed) | T | N/A | 12, 14, 15, 19, 20, 26, 15 | Normal hardware-firmware flow |
| Get UDID (directed) | T | N/A | 12, 14, 16, 19, 20, 26, 27, 29 | The hardware returns its UDID and corresponding slave address. |
| None of the above ARP protocols | N/A | N/A | 12, 14, 16, 19, 20, 26, 27, 28 | Normal hardware-firmware flow (error handling situation) |

The hardware will ACK all valid transactions above. An exception is when processing the assign address protocol, if the received UDID does not match any internal UDID the hardware begins NACKing after the first unmatched byte. Another exception is that if the address resolved flag is asserted and a Get UDID (general) command is received, the command byte is NACKed. Finally, a PEC failure causes a transaction to be NACKed.

In both directed protocols, Get UDID (directed) and reset device (directed), the second byte, or ARP command byte, consists of the target slave address with an appended 0 or 1 to denote a reset device or Get UDID protocol, respectively. Therefore, the hardware uses address bits [7:1] for address comparison and LSB bit 0 to determine the protocol.

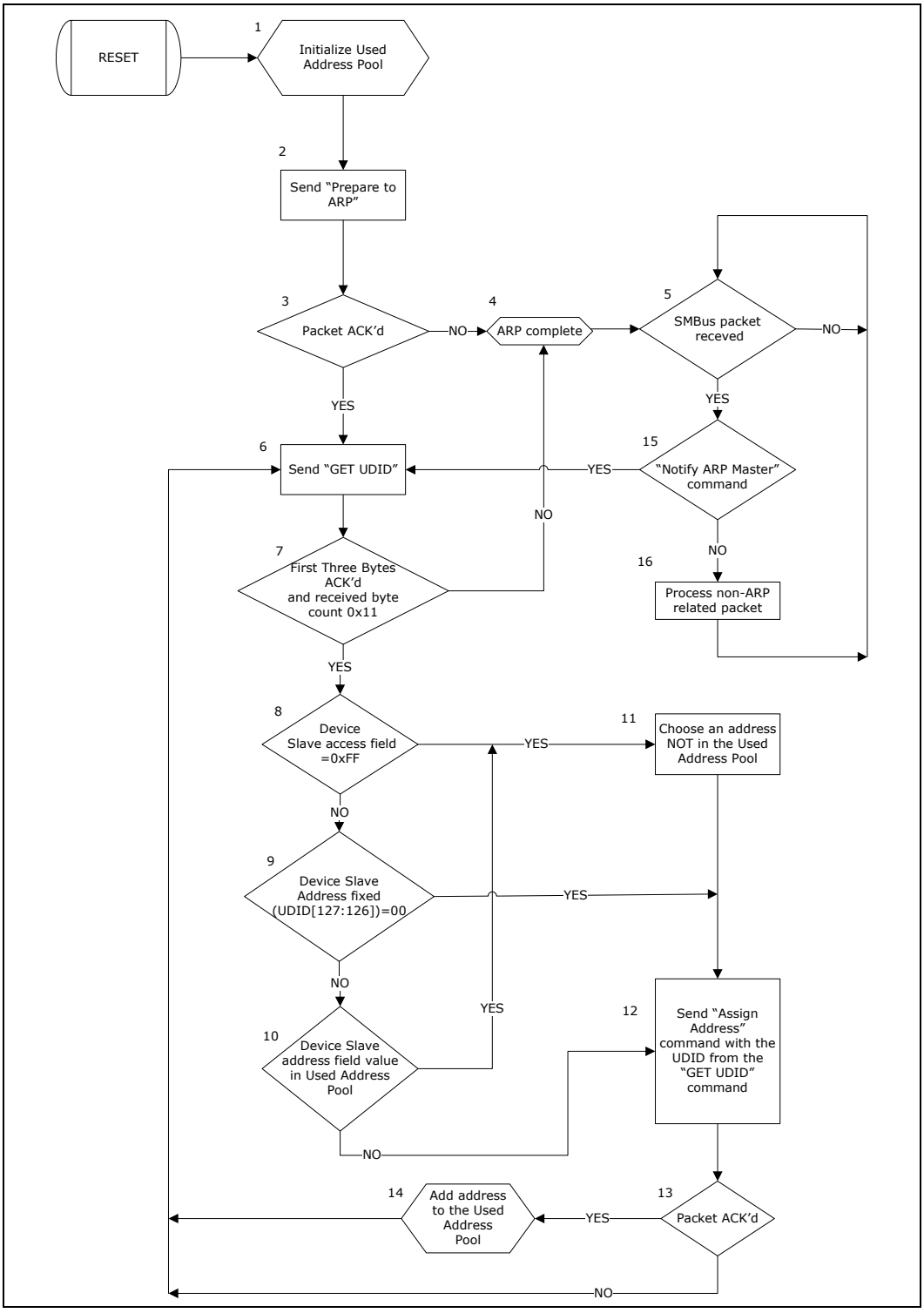
Since the SMT supports multiple UDIDs, during the ARP process it participates in the Get UDID (general) and assign address protocols once for each UDID such that each UDID has a slave address which is resolved by the ARP master.

For GPBR and all both Get UDID flavors, the transaction is descriptor-only with no target data written to memory, and the descriptor is managed by the hardware.

*Note:* The firmware implements a time-out mechanism such that if the SoC issues the Notify ARP Master command and the ARP master does not respond within a particular time period, then the SoC re-issues the Notify ARP Master command. It is implemented to comply with the *System Management Bus (SMBus) Specification, Version 3.0* for bus timing.

### 29.4.2.4 ARP Master Behavior

**Figure 29-3. ARP Master Behavior Flow Diagram**



The ARP master must always execute the ARP when it enters the working state and anytime it receives an SMBus status change indication (due to device addition or removal). The process begins with the ARP master issuing the Prepare To ARP command. In all cases the ARP master must be able to resolve addresses when it receives the Notify ARP Master command.

Since SMBus devices join the system without a corresponding system reset (i.e., a hot-plug event), the ARP master optionally chooses to issue the Get UDID (general) command at least once every 10 seconds to discover newly added devices that require address resolution but which do not support the Notify ARP Master command. No device whose AR flag is set responds to this command. However, a newly added device enters the system with a power-up reset, which resets its AR flag; it responds to a Get UDID (general) command with its UDID. The host chooses to assign such a newly added device a non-conflicting address or chooses to re-ARP the entire bus.

Until the ARP process is complete, the ARP master must not wait more than 2 seconds before issuing a Get UDID (general) command after issuing the previous Get UDID (general) command. This restriction is important to allow another SMBus master to determine when it is safe to do an enumeration of the bus.

**Table 29-10. ARP Master Operation (Sheet 1 of 2)**

| Step | | Description |
|---|---|---|
| 0. | Perform ARP master initialization. | • [FW] See Section 29.4.2.5, "ARP Initialization Flow" on page 746. |
| 1. | Upon starting, the ARP master initializes its used address pool. Initially this consists of the slave addresses of fixed-address SMBus devices known to the ARP master and reserved addresses (as defined in *SMBus 2.0 Specification*). | • [FW] |
| 2. | Send the Prepare To ARP command. | • [FW] |
| 3. | Check for an acknowledgment for all bytes in the previous packet. If any bytes were not acknowledged then the ARP master assumes that no ARP-capable devices are present and therefore consider the ARP process complete; proceed to step 4. If all bytes were acknowledged then go to step 6. | • [FW] |
| 4. | The ARP master found no response to the Prepare To ARP command so it assumes that no ARP-capable devices are present in the system at this time. The ARP master periodically re-issues the Prepare To ARP command to discover any ARP-capable devices added. Proceed to step 5. | • [FW] |
| 5. | Wait for an SMBus packet. If a packet is received proceed to step 15. | • [FW] |
| 6. | Send the Get UDID command. | • [FW] |
| 7. | Check for an acknowledgment for the first three bytes and verify that the byte count value received is 0x11. If not, then the ARP master assumes that an ARP-capable device(s) is no longer present and therefore consider the ARP process complete; proceed to step 4. Otherwise proceed to step 8. | • [HW and FW] HW verifies the byte count before sending descriptor to FW. |
| 8. | Check the value of the device slave address received. If 0xFF then proceed to step 11 since this device does not possess a valid slave address. Otherwise proceed to step 9. | • [FW] |
| 9. | Determine if this device has a fixed slave address. If bits 127 and 126 of the UDID are 00b then it has a fixed address, so proceed to step 12. Otherwise proceed to step 10. | • [FW] |
| 10. | The device possesses a valid slave address. However, the ARP master must check this address against the used address pool to ensure that no other device has already been assigned the same address. If the received device slave address is found in the used address pool then proceed to step 11. If not, then the device keeps its current slave address but needs acknowledgment from the ARP master; proceed to step 12. | • [FW] |

### Table 29-10. ARP Master Operation (Sheet 2 of 2)

| Step | | Description |
|---|---|---|
| 11. | Select a slave address that is not in the used address pool and proceed to step 12. | • [FW] |
| 12. | Send the Assign Address command with the UDID returned by the device in the Get UDID command packet. | • [FW] |
| 13. | Check for acknowledgment of all bytes in the Assign Address command packet. If any byte was not acknowledged then the ARP master assumes the device is no longer present; proceed to step 6 to determine if more devices require address resolution. If all bytes were acknowledged then the ARP master assumes that the device has accepted the address assignment; proceed to step 14. | • [HW and FW] HW monitors slave ACK before sending descriptor to FW. |
| 14. | The device now has a valid slave address. The ARP master must add this address to the used address pool. Proceed to step 6 to determine if more devices require address resolution. | • [FW] |
| 15. | The ARP master checks if the received packet was the Notify ARP Master command. If so, then it must execute the ARP to resolve the address for the newly added device(s); proceed to step 6. If not, then proceed to step 16. | • [FW] |
| 16. | The ARP master received a non-ARP related packet. Process it accordingly and proceed to step 5. | • [FW] |

These steps cover the case when the ARP master has exited a reset state. Since ARP supports hot-plug, the master must be prepared to execute the ARP at any time; step 15 covers the case when a device issues the Notify ARP Master command. If during the ARP process, a previously detected device is now not discovered, then the slave address associated with that device is removed from the used address pool.

The diagram does not consider bus time-out mechanisms or retries. These are implemented to comply with SMBus timing requirements.

### 29.4.2.5 ARP Initialization Flow

The firmware initializes the SoC for ARP protocols before any SMBus traffic. Because the SMT master behavior applies to both master-transmitter (e.g., ARP host issuing a prepare to ARP) and slave-transmitter (e.g., ARP slave issuing a host notify), and SMT target behavior applies to both master-receiver (e.g., ARP host receiving a host notify) and slave-receiver (e.g., ARP slave receiving a prepare to ARP), many ARP initialization steps are common to host and slave devices.

The exception is UDID assignment. This pertains only to ARP slaves.

**Table 29-11. ARP Initialization Flow (Sheet 1 of 2)**

| Initialization Flow for an ARP Slave | Initialization Flow for the ARP Host |
|---|---|
| Configure the SoC bootstrap configuration information programmed fields.<br>• Preset UDID fields are loaded from the SoC bootstrap configuration information.<br>• Also, the SoC bootstrap configuration information is loaded to configure the unique UDID field (VSID)—the hardware permutes the SoC bootstrap configuration information to program multiple UDIDs as necessary.<br>***Example:*** To support two functions each with two UDIDs, combine three hard-coded LSBs with a sufficient number of SoC bootstrap configuration information bits (n) to distinguish all SoC devices.<br>**Bit Fields:**<br>VSID = [31:32-*n*] [32-*n*-1:3] [2:1] [0]<br>VSID = SoC bootstrap configuration information bits static-bits function UDID<br>Fn0/UDID0 = {SoC bootstrap configuration information bits}{static bits}{00}{0}<br>Fn0/UDID1 = {SoC bootstrap configuration information bits}{static bits}{00}{1}<br>...<br>Fn1/UDID1 = {SoC bootstrap configuration information bits}{static bits}{01}{1}<br>The SoC bootstrap configuration information bits are shared by all functions and UDIDs.<br>• Preset other fields based on the SoC bootstrap configuration information: GPBRCTRL.GPTRADR, SPGT.SPD, TPOLICY.ADDR0_EN, TPOLICY.ADDR1_EN, and TPOLICY.Host_SMBADDR_EN. | Configure the SoC bootstrap configuration information programmed fields.<br>• N/A<br>• N/A<br>• Preset other fields based on the SoC bootstrap configuration information: GPBRCTRL.GPTRADR, SPGT.SPD, TPOLICY.ADDR0_EN, TPOLICY.ADDR1_EN, and TPOLICY.Host_SMBADDR_EN. |
| Program SMBus address to any fixed-address target.<br>• Program TACTRL.ADDR0 if required based on UDID0.DEVCAP field.<br>• Program TACTRL.ADDR1 if required based on UDID1.DEVCAP field. | |
| Verify transaction status flags (no invalid state flags).<br>• Verify all 0s in fields TSTS.IP and MSTS.IP.<br>• Likewise, verify numerous error-status registers. | |
| Program descriptor base address and buffer entries.<br>• Program MDBA and MDS.<br>• Program TBBA and TBS.<br>• Program SMTICL. | |
| Program head and/or tail pointers.<br>• Program MCTRL.FMHP and MSTS.HMTP.<br>• Program HTHP.HTBHP and FTTP.FTBTP. | |

### Table 29-11. ARP Initialization Flow (Sheet 2 of 2)

| Initialization Flow for an ARP Slave | Initialization Flow for the ARP Host |
|---|---|
| Enable (unmask) interrupts from the hardware due to certain conditions.<br>• Assert/de-assert ERRINTMSK, ERRAERMSK, TCTRL.TIE, MCTRL.MEIE. | |
| Configure policies.<br>• Program RPOLICY.TBRCLKCNT, RPOLICY.COLRTRY, RPOLICY.TBRBCLK, RPOLICY.RETRY. | |
| Configure GPBR<br>• Set GPBRCTRL.EN.<br>• Assert/de-assert GPBRCTRL.PECEN.<br>• Assert/de-assert GPBRCTRL.HWCLRDIS.<br>• Program GPBRCTRL.BC.<br>• Program GPBRCTRL.CMD.<br>• Program GPBRCTRL.GPTRADR. | |

### 29.4.3 SMT System Usage Models

The SMT architecture takes into account various usage models as envisioned on the SoC-based platforms. The focus of the architecture is to keep the hardware overhead low for supporting the various usage models and protocols:

- SMBus ARP mastering or ARP target
- Embedded Controller (EC) on the SMBus communicating with the SoC

### 29.4.4 SMT Security Requirements

The SMT has no security requirements.

### 29.4.5 SMT Timing Modes

The SMT currently supports three different timing modes: standard (up to 100 kHz), fast-mode (up to 400 kHz), and fast-mode plus (up to 1 MHz). The timing requirements for fast-mode and fast-mode plus are found in the $I^2C$ industry specification.

The following section covers design targets for the various timing modes in various operating conditions. The SMT controller does not contain dynamic timing adjustment to account for varying bus loads or pull-up resistor selections, thus SMT is designed and have default settings to meet the specification in the worst-case scenarios. This comes at the cost of not having optimal frequency for any given timing mode in nominal conditions.

*Note:* Devices also stretch the clock low thus reducing the frequency. These timing estimates are assuming the device is not further impacting the timings.

**Table 29-12. SMT Timing Mode Maximum Clock Frequency Ranges**

| Mode | Maximum Clock Frequency Range |
|---|---|
| Standard | 90-100 kHz* |
| Fast Mode | 350-400 kHz* |
| Fast Mode Plus | 750-1000 kHz* |

*Timing assumes the platform design is within the specification for maximum rise time allowed requiring appropriate Rpullup value selection for the given capacitive load on the bus.

## 29.4.6 SMT as Master

The SMT as the initiator provides the hardware for the internal agents inside the SoC to send/receive data across the SMBus. Due to the various usage models that SMT supports, the hardware support exists for initiating reads/writes from/to external devices on the SMBus. The SMT has the hardware capability to transport them over the physical SMBus and report the status back to the firmware.

*Note:* The firmware must ensure the data in the transmit data buffers is arranged in the order that it wants to send on the bus, i.e., data pointed to by the Tx data pointer are sent first on the SMBus, then the next byte, etc.

With respect to the receive buffer, data are written to memory as-is received from the physical SMBus, i.e., the first byte received is placed in the lowest address, then the next byte, etc.

### 29.4.6.1 Hardware Buffering for Master Support

The hardware implements a 240-byte buffer which it uses to queue transactions (both master-transmitter and master-receiver) as a descriptor-based master. Though the hardware supports a maximum read/write of 240 bytes, the firmware driver restricts the payloads to <=32B to compliance SMBUS spec. Please see Section 29.4.6.3.2, “PEC-Enabled SMBus Transactions” for details.

The storage queues are not shared between the master and the target. Both logical sides function independently without dependency on each other.

*Note:* Although SMT theoretically supports master-initiated write and read cycles of arbitrary length, the practical expectation is that the transaction size does not exceed 32B with Intel firmware driver.

*Note:* When SMT as master sends I2C write or read, the hardware can support a maximum payload 240 bytes with custom external driver.

### 29.4.6.2 Master Descriptor

The master descriptor is a ring buffer of individual descriptors set up by the firmware as shown in Figure 29-4. The descriptor ring buffer is pointed to by the base (MDBA) and the size is indicated by the MD Size register (MDS). See register definitions for details. Individual descriptors have a 64-bit pointer into the data buffer with an expected transmit and/or receive length programmed in the control field of the descriptors.

The hardware updates the status WB field in Figure 29-4 and Figure 29-5. See Table 29-13, "Master Descriptor Field Descriptions" for detailed descriptions. The remaining three Dwords are updated by the firmware.

The firmware always leads and points by the FWmHeadPtr (MCTRL.FMHP). The hardware always points by the HWmTailPtr (MSTS.HMTP).

**Figure 29-4. Master Descriptor Ring Buffer**

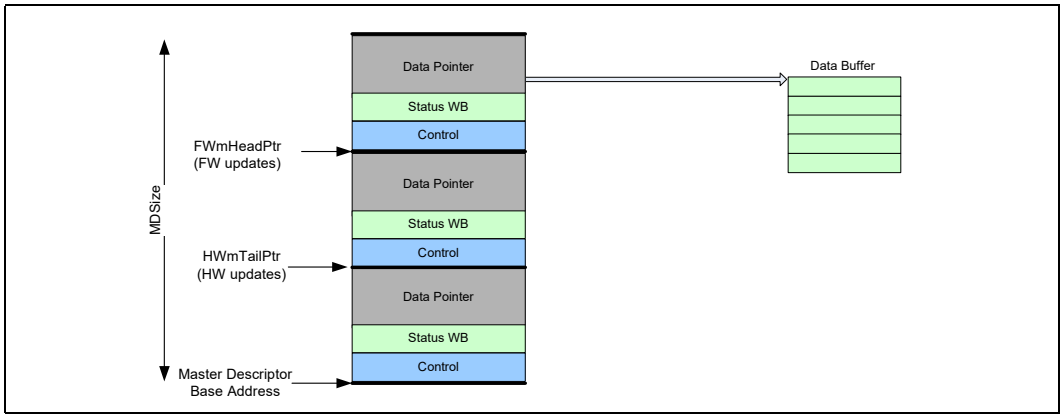
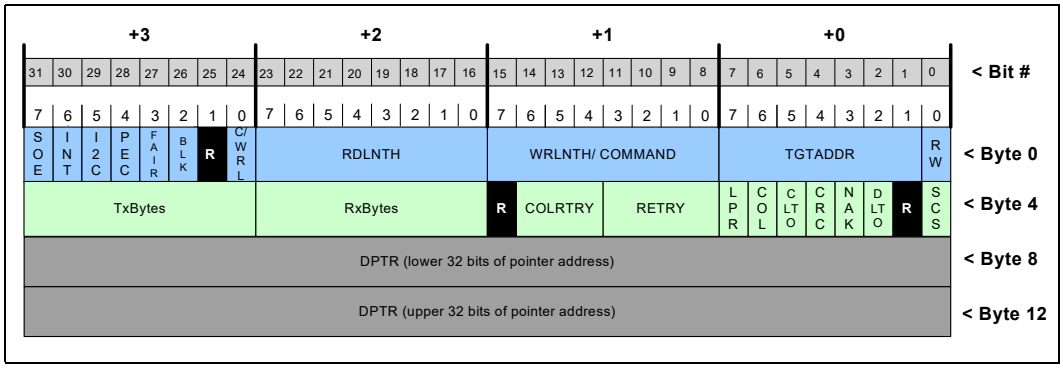



**Figure 29-5. Master Descriptor Format**



## Table 29-13. Master Descriptor Field Descriptions (Sheet 1 of 3)

| Dword # | Bit # | Field | Description |
|---|---|---|---|
| 0 | 31 | SOE | **Stop On Error**: This bit is set to 1 to direct the hardware that if a descriptor-based master cycle results in an unsuccessful transaction on SMBus, the hardware must clear the Start/Stop bit (MCTRL.SS) and stop the engine.<br>If this bit is clear if the hardware encounters an error while sending a transaction on SMBus, it sets the Master Error Status bit (MSTS.MEIS) and continues to process the next descriptor in the queue. |
| 0 | 30 | INT | **Interrupt**: This bit is set to 1 to direct the hardware that it must generate an MSI to the firmware when it has completed the requested transaction successfully. This interrupt is generated only after the status Dword of the descriptor is written back to memory.<br>Interrupts due to unsuccessful transactions on the SMBus are not affected by this bit.<br>If this bit is clear and the transaction is successful on the SMBus, no interrupt is generated; however, the firmware polls the MSTS.MIS bit.<br>***Note:*** When using legacy interrupts, this bit has no effect; a legacy interrupt is always generated even if not requested by the master descriptor (firmware). |
| 0 | 29 | $I^2C$ | $I^2C$ **Enable:** This bit is set to 1 to indicate the hardware must perform the transaction using $I^2C$ protocol. |
| 0 | 28 | PEC | **Packet Error Code**: This bit is set to 1 to indicate the hardware must append CRC (or PEC) as the master of the requested transaction if it is a write or check received CRC (PEC) of the requested transaction if it is a read. PEC is calculated over the entire message (including address and read/write bits) and supplied by the device which puts out the last-data byte of the message. |
| 0 | 27 | FAIR | **Fair**: This bit is set to 1 to indicate if the hardware is able to successfully win arbitration on the bus and master the transaction, it must set its internal fairness flag.<br>This allows a mechanism for fairness on the SMBus per *MCTP Specification*. |
| 0 | 26 | BLK | **Block**: Set to 1 by the firmware to indicate the hardware must perform a block transaction on the bus. The hardware determines one of three block transactions based on the following fields<br>{BLK, C/WRL and R/W}.<br>100: Perform an SMBus block write.<br>111: Perform an SMBus block read.<br>101: Perform an SMBus block process call.<br>Others: Reserved |
| 0 | 25 | Reserved | Reserved |
| 0 | 24 | C/WRL | **Command/Write Length**: Set to 1 by the firmware when it has overloaded the write length field with the command code of the SMBus transaction. |
| 0 | 23:16 | RDLNTH | **Read Length**: Indicates the number of bytes the hardware receives from the target, and writes to the receive data buffer one-based counting, i.e., the value 0h means 0 bytes of receive data and the value Ah means 10 bytes of receive data. The maximum read length currently supported by the hardware is 240 bytes due to its internal buffer size. |
| 0 | 15:8 | WRLNTH | **Write Length**: Indicates the number of bytes the hardware transmits as a master (except the first address byte, byte count, and any subsequent address bytes for reads) one-based counting, i.e., the value 0h means 0 bytes of transmit data and the value Ah means 10 bytes of transmit data. The maximum write length supported by the hardware is 240 bytes including address due to the size of its internal buffer. The byte count for the SMBus transactions is calculated by the hardware based on the write length field. Example: If WRLNTH = 8, the hardware calculates byte count = 7 and sends on the SMBus. The WRLNTH field itself contains command code and 7 bytes of data. When the C/WRL flag is set, this field contains the command code of the SMBus transaction. |

## Table 29-13. Master Descriptor Field Descriptions (Sheet 2 of 3)

| Dword # | Bit # | Field | Description |
|---|---|---|---|
| 0 | 7:1 | TGTADDR | **Target Address**: 7-bit address field indicating the target SMBus/$I^2C$ address.<br>***Note:*** See Section 29.4.6.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 764 for restrictions on writes to certain addresses. |
| 0 | 0 | RW | **Read/Write**: Set to 1 to indicate a read request. Cleared to 0 to indicate a write request. Combinations of this bit and the WRLNTH, RDLNTH, and C/WRL fields are decoded by the hardware to distinguish between various types of SMBus cycles and $I^2C$ cycles.<br>***Note:*** For SMBus Process Call commands, this bit must always be cleared to 0.<br>***Note:*** See Section 29.4.6.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 764 for restrictions on writes to certain addresses. |
| 1 | 31:24 | TxBytes | **Transmitted Bytes**: The hardware updates this field to indicate how many bytes transmitted by it were ACKed by the target. This field provides the firmware the ability to reconstruct which particular byte was NACKed by the target.<br>This count is one-based and includes all the bytes sent by the hardware which are ACKed by the target including the address.<br>Value of 0 means address phase NACKed or collision on the address phase. |
| 1 | 23:16 | RXBytes | **Received Bytes**: The hardware indicates how many bytes of received data it is writing to memory (into the data buffer). The count is one-based, i.e., the value 0h means 0 bytes of data. The hardware limitation is for a 240-bytes data buffer implying that for a maximum SMBus legal block read of 32 bytes with the PEC, it forwards to memory the 240 bytes of data; the byte count received.<br>***Note:*** PEC is not forwarded to the firmware, since the firmware explicitly enables/disables PEC per master transaction. The hardware returns the PEC check in the CRC field, which the firmware inspects. |
| 1 | 15 | Reserved | Reserved |
| 1 | 14:12 | COLRTRY | **Collision Retry**: The hardware indicates the number of collisions on the last attempt before retiring the descriptor. |
| 1 | 11:8 | RETRY | **Retry Count**: The hardware indicates the count with the number of retries attempted before retiring the descriptor. |
| 1 | 7 | LPR | **Large Packet Received**: The hardware sets this bit to indicate that more data was sent by the target than expected by the firmware and exceeds the allocated receive data space (TRxCTRL.MRxB).<br>The hardware must DMA the data to the buffer in memory in the space allocated, dropping the extra bytes. |
| 1 | 6 | COL | **Collisions**: Set to 1 by the hardware to indicate that failure was due to number of collisions exceeding the collision retry count (RPOLICY.COLRTRY). |
| 1 | 5 | CLTO | **Clock Low Time Out**: Set to 1 by the hardware to indicate unexpected time-out seen on the bus during the course of the request.<br>This bit being set indicates that SMB clock signal was held low by the external device for the count as programmed in CNT. |
| 1 | 4 | CRC | **CRC Error:** Set to 1 by the hardware to indicate CRC error[1] on the request.<br>For read requests with PEC, the hardware sets this bit if the PEC received from the target does not match the PEC calculated by the hardware.<br>For write requests with PEC, the hardware sets this bit if the target NACKs the PEC byte. |
| 1 | 3 | NAK | **NACK Received:** Set to 1 by the hardware to indicate unexpected NACK asserted by the target. |
| 1 | 2:0 | Reserved | Reserved |
| 1 | 0 | SCS | **Success**: Set to 1 by the hardware to indicate that the cycle was transferred successfully. Zero indicates that some error was encountered and other bits in the status indicate the error. |

## Table 29-13. Master Descriptor Field Descriptions (Sheet 3 of 3)

| Dword # | Bit # | Field | Description |
|---|---|---|---|
| 2 | 31:0 | DPTR | **Data Pointer:** Byte-aligned pointer to the starting location of the data buffer.<br>The hardware reads from memory and transmits exactly as many bytes as is indicated by the WRLNTH field. This does not include the address fields of the message (i.e., start address/repeated start address) and byte count if any.<br>The hardware is expected to write to memory exactly as many bytes as is indicated by the RDLNTH field (unless an error condition exists and the target fails to provide the expected number of bytes, in which case the appropriate error bits are set). |
| 3 | 63:32 | DPTR | |

1. The *System Management Bus (SMBus) Specification, Version 3.0* is vague about the condition of the CRC error when the master is initiating a READ and the target provides the CRC. Irrespective of the fact that the CRC is correct or incorrect, the master must NACK the cycle and assert STOP. The SMT hardware informs the firmware of the incorrectness of the received CRC and leaves it up to the firmware for further action. See the *System Management Bus (SMBus) Specification, Version 3.0*.

### 29.4.6.3 Master Descriptor Usage

Descriptors are used for SMT transactions the are initiated by the firmware. The hardware decodes the $I^2C$, PEC, WRLNTH, and RDLNTH fields to differentiate between the types of SMBus transactions (see Section 29.4.6.3.1, "SMBus Protocol Transfers" on page 754) and $I^2C$ transactions (see Section 29.4.6.3.3, "$I^2C$ Protocol Transfers" on page 757).

The master transaction flow is defined in Section 29.4.6.4, "Master Transactions Flow" on page 758.

The master retry flow is defined in Section 29.4.6.6, "Master Retry Flow" on page 762.

#### 29.4.6.3.1 SMBus Protocol Transfers

Table 29-14 illustrates how the hardware uses the control information in the descriptor.

*Note:*

1. The firmware must ensure that the $I^2C$ bit is 0 when initiating SMBus transactions.
2. Target address and R/W# indication is taken from the descriptor.
3. PEC-enabled write transactions proceed in the same manner except that the hardware appends PEC after the last data phase.
4. PEC-enabled read transactions instruct the hardware to check CRC and report the status to the firmware.

*Note:* See Section 29.4.6.7, "Write Disabling to DIMM SPD EEPROM Addresses" on page 764 for restrictions on writes to certain addresses.

**Table 29-14. SMBus Transaction Encodings (Sheet 1 of 2)**

| SMBus Command | BLK | C/WRL | WRLNTH | RDLNTH | RW (0=W; 1=R) | DPTR (Points to TX Data) | DPTR (Points to RX Data) |
|---|---|---|---|---|---|---|---|
| Quick Command | 0 | 0 | 0 | 0 | Read/ Write | X | X |
| Send Byte | 0 | 1 | Command | 0 | 0 | X | X |
| Receive Byte | 0 | 0 | 0 | 1 | 1 | X | Valid and points to a buffer where the received byte is placed. |
| Write Byte | 0 | 0 | 2 | 0 | 0 | Valid and points to 2 bytes (command and 1-data byte). | X |
| Write Word | 0 | 0 | 3 | 0 | 0 | Valid and points to 3 bytes (command and 2-data bytes). | X |
| Read Byte | 0 | 1 | Command | 1 | 1 | X | Valid and points to a buffer where the received byte is placed. |
| Read Word | 0 | 1 | Command | 2 | 1 | X | Valid and points to a buffer where the received 2 bytes are placed. |
| Process Call | 0 | 0 | 3 | 2 | 0 | Valid and points to 3 bytes (command and 2-data bytes). | Valid and points to a buffer where the received 2 bytes are placed. |

## Table 29-14. SMBus Transaction Encodings (Sheet 2 of 2)

| SMBus Command | BLK | C/WRL | WRLNTH | RDLNTH | RW (0=W; 1=R) | DPTR (Points to TX Data) | DPTR (Points to RX Data) |
|---|---|---|---|---|---|---|---|
| Block Write (1 byte) | 1 | 0 | 2 | 0 | 0 | Valid and points to 2 bytes (command and 1-data byte); the byte count is calculated by the hardware. | X |
| Block Write (2 bytes) | 1 | 0 | 3 | 0 | 0 | Valid and points to 3 bytes (command and 2-data bytes); the byte count is calculated by the hardware. | X |
| Block Write (3 bytes or more) | 1 | 0 | 4 or more[1] | 0 | 0 | Valid and points to at least 4 bytes (command and 3-data bytes); the byte count is calculated by the hardware. | X |
| Block Read (1 byte) | 1 | 1 | Command | 2 | 1 | X | Valid and points to a buffer where the received 2 bytes are placed (byte count, data byte 1). |
| Block Read (2 bytes or more) | | 1 | Command | 3 or more[2] | 1 | X | Valid and points to a buffer where at least the received 3 bytes are placed (byte count, data byte 1, etc.). |
| Block Write-Block Read Process Call[3] (write 1 byte, read N) | 1 | 0 | 2 | N[4]+1 | 1 | Valid and points to 2 bytes (command and 1-data byte); the byte count is calculated by the hardware. | Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes). |
| Block Write-Block Read Process Call (write 2 bytes, read N) | 1 | 0 | 3 | N+1 | 1 | Valid and points to 3 bytes (command and 2-data bytes); the byte count is calculated by the hardware. | Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes). |
| Block Write-Block Read Process Call (write >2 bytes, read N) | 1 | 0 | 4 or more | N+1 | 1 | Valid and points to at least 4 bytes (command and 3-data bytes); the byte count is calculated by the hardware. | Valid and points to a buffer where at least the received (N+1) bytes are placed (byte count, N data bytes). |

1. Per the *System Management Bus (SMBus) Specification, Version 3.0*, having a block write of exactly 3 bytes can occur. However, a block write of 3 bytes and a write word have identical signaling on the bus.
2. Per the *System Management Bus (SMBus) Specification, Version 3.0*, having a block read of exactly 2 bytes can occur. However, a block read of 1 data byte (address, command, byte count = 1, followed by 1 data byte) and a read word from the target (address, command, DataByte1, DataByte2) have identical signaling on the bus.
3. The sum of the data bytes in the write and read phases must not exceed 32 bytes per the *System Management Bus (SMBus) Specification, Version 3.0*.
4. N must be greater than 1.

#### 29.4.6.3.2 PEC-Enabled SMBus Transactions

For writes going out from the SMT to the SMBus, the PEC is calculated by the hardware and appended. For reads, the PEC byte is received from the target and verified by the hardware (incorrectness notification is provided to the firmware by the status bits). The PEC byte is not sent to the buffer.

*Note:* Firmware-specific:

1. If the firmware wants to send a large amount of data (more than 32 data bytes without counting command byte and byte count) to a target using the SMBus 2.0 protocol and indicates to the hardware to send data using block write protocol, the firmware is responsible to break the data into multiple packets and set up individual descriptors for each packet.
2. The firmware honors all SMBus 2.0 rules for transactions of write, read, transcribe of the total data transfer on the SMBus being not more than 32 bytes (including BlockWr-BlockRd process call).
3. The firmware sets the PEC bit when initiating a PEC-accompanied read from a target so that the hardware verifies and reports PEC accuracy, or when initiating a PEC-accompanied write transaction so that the hardware automatically appends the PEC byte.
4. The firmware must clear the $I^2C$ bit when initiating the SMBus transactions.
5. Because a single pointer is used, for SMBus transactions which both transmit and receive data (i.e., process call and block write-block read process call), the received data overwrites the transmitted data. Before beginning the transaction, the firmware must copy the transmit data if the transmit data are preserved after the transaction.

#### 29.4.6.3.3 I²C Protocol Transfers

For I²C transactions, Table 29-15 illustrates how the hardware uses the control information in the descriptor. The following are the I²C support limitations:

1. Support is limited to 7-bit addressing mode only.
2. Write-read combined format is supported with SMT as a master.
3. As a target, the hardware supports only writes initiated by an external master.
4. When the SMT sends/receives I2C write/read, Hardware supports payloads any size up to 240B.

**Table 29-15. I²C Commands**

| I²C Command | C/WRL | WRLNTH | RDLNTH | RW (0=W; 1=R) | DPTR (points to TX data) | DPTR (points to RX data) |
|---|---|---|---|---|---|---|
| I²C Writes (MTx-to-SRx) | 1 | Command (1 byte of write data) | 0 | 0 | X | X |
| I²C Writes (MTx-to-SRx) | 0 | 2 or more | 0 | 0 | Points to the transmit buffer containing at least 2 bytes of write data. | X |
| I²C Reads (STx-to-MRx) | 0 | 0 | 1 or more | 1 | X | Points to the receive buffer where the receive data are placed. |
| I²C Combined Format (write followed by read) | 1 | Command (1 byte of write data) | 1 or more | 0 | X | Points to the receive buffer where the receive data are placed. |
| I²C Combined Format (write followed by read) | 0 | 2 or more | 1 or more | 0 | Points to the transmit buffer containing at least 2 bytes of write data | Points to the receive buffer where the receive data are placed. |

***Notes:***
1. This table assumes the target address is programmed in the descriptor and is not part of the WRLNTH field.
2. The I²C bit must be set for all I²C transactions.
3. PEC is not supported for any I²C transaction.
4. The BLK bit must be 0 for all I²C transactions.

### 29.4.6.4 Master Transactions Flow

All master transactions on the physical SMBus pins use descriptors. The high-level flow typically is:

1. The firmware sets the data structures in memory.
2. The firmware programs the descriptors and the associated hardware.
3. The firmware sets the Start bit to initiate the transactions.
4. The hardware processes the descriptor, first setting the InProgress bit, and completes the transaction.
5. The hardware writes data back to memory (if any).
6. The hardware writes back the status to memory for the processed descriptor.
7. If no more descriptors are processed, the hardware clears the InProgress bit.

#### 29.4.6.4.1 Firmware Assumptions

The assumption is an SMT firmware driver exists that understands the SMT hardware register interface and the usage for sending and receiving SMBus messages.

#### 29.4.6.4.2 Initialization

1. The firmware allocates a 64B aligned buffer in memory to be used as the master descriptor ring buffer. Each descriptor is 16B long.
2. The firmware then programs up the MD Base Address (MDBA) register with the lower memory address of the descriptor ring buffer, and MD Size (MDS) register containing the number of descriptors in the ring.
3. The firmware initializes the FWmHeadPtr (MCTRL.FMHP) and HWmTailPtr (MSTS.HMTP) by writing all 0 into it.
4. The firmware also programs interrupts as needed.

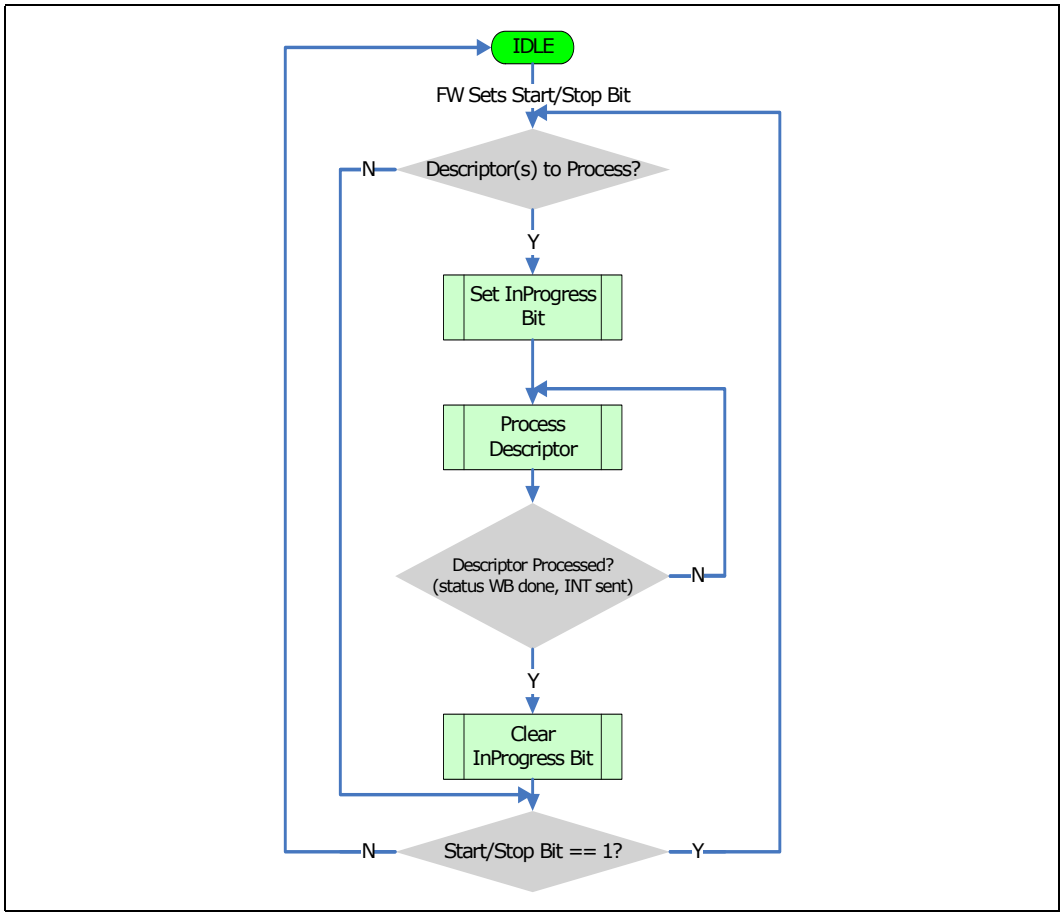
*Note:* The firmware schedules a master transaction in the descriptor ring buffer if the buffer is not full. The descriptor ring buffer is full if (FWmHeadPtr == HWmTailPtr - 1). The –1 subtraction here needs to account for buffer utilization of N-1 for an N-deep buffer.

#### 29.4.6.4.3 Hardware-Firmware Flow

When the firmware driver has to initiate a transaction on the SMBus, and the descriptor ring buffer is not full, the following steps are performed:

1. The firmware programs a 16B descriptor with the attributes of the transaction in memory.
2. The firmware increments the FWmHeadPtr register.
3. The firmware then sets the Start bit (MCTRL.SS) in the hardware.
4. The hardware continuously checks if a descriptor needs to be processed by checking (FWmHeadPtr != HWmTailPtr):
   a. Buffer empty condition: MSTS.HMTP = MCTRL.FMHP
   b. Buffer full: MCTRL.FMHP = MSTS.HMTP-1 or MCTRL.FMHP-MSTS.HMTP = MDS
   c. Buffer wrap condition: When MSTS.HMTP = MDS, and MCTRL.FMHP has already wrapped around, i.e., >= 00h, the hardware reads the 16B descriptor, processes it and if the criteria to increment the pointer are met, it wraps to 00h.
5. If a descriptor (see Section 29-6, "Hardware-Firmware Flow Diagram—DMA Mode" on page 760) is available, the hardware first sets the InProgress bit (MSTS.IP), and then reads the 16B descriptor from memory by combining (MD Base + HWmTailPtr) from which it:
   a. Decodes the control Dword for transaction type and other attributes.
   b. Loads the write data and read data pointers as required.
   c. For writes, the hardware fetches data from the memory pointed to by the DPTR and transmits on the wire. For reads, the hardware stores the DPTR pointer so it can Direct Memory Access (DMA) the data to that address when data is provided by the target.
6. Once the transaction is completed, the hardware does a status write back to the status WB Dword.
7. The hardware then increments HWmTailPtr and sends the interrupt to the firmware if enabled to do so.
   a. Before issuing the MSI, the hardware denotes the master completion in SMTICL and any error status.
8. The hardware then clears the InProgress bit to indicate it has completed processing of the descriptor.
9. The hardware then checks if the Start bit is set. If so, flow follows from #4, or else from #1.

**Figure 29-6. Hardware-Firmware Flow Diagram—DMA Mode**



### 29.4.6.5 Clearing of the Start Bit

Under certain conditions, the hardware clears the Start bit (MCTRL.SS).

The hardware clears this register for the following fatal error scenarios:

- The transaction is unsuccessful on the final retry (if enabled), including collisions.
- The hardware receives unsuccessful completion on the internal SoC I/O fabric.

The hardware clears this register for the following non-fatal scenario:

- Master descriptor Stop-on-Error (SOE) bit was set and the transaction was unsuccessful.

The firmware clears the Start bit when it kills a master transaction:

1. The firmware clears MCTRL.SS.
2. The firmware sets GCTRL.KILL.
3. The firmware polls MSTS.IP; the transaction is killed when the bit is detected as cleared.

### 29.4.6.6 Master Retry Flow

The hardware contains separate timers and counters to auto-retry unsuccessful SMBus cycles that it masters on the SMBus.

Two counters for time between retries and retries due to collision are on the bus. Any failed cycle due to a collision first exhausts the collision counter before decrementing the retry counter. The time between retries is maintained only between two successive transaction retries and not between two successive retries due to a collision.

The high-level flow in the pseudo code is:

```
//Master Transaction begin

MASTER_FLOW

// All registers are defined in RPOLICY register

RELOAD RETRY

// Collision retries

COLRTRY

// Time between retries

TBR

While (RETRY >= 0) {

    RELOAD TBR

    RELOAD COLRTRY

    Send Cycle on SMBus

    While (SMBus COLLISION & (COLRTRY > 0)) {

        COLRTRY = COLRTRY - 1

        WAIT for SMBus IDLE

        Send Cycle on SMBus

    }

    If (SMBus ACK) {

    // Successful cycle

    Update Status in descriptor and WB to memory

    Send Interrupt if enabled

    GOTO MASTERFLOW and wait for new cycle

        BREAK

    } else if

    // Unsuccessful cycle

    (((RETRY == 0) & SMBus NACK) || SMBus Timeout) {

    Update Status in descriptor and WB to memory
```

```
    Set Master Error and send MSI (if enabled) after updating SMTICL

    Clear DMA Start/Stop Bit

GOTO MASTERFLOW and wait for FW to set Start bit

    BREAK

    } else if (SMBus NACK & (RETRY > 0)){

        RETRY = RETRY - 1

        CNTDOWN TBR

        WAIT for TBR to expire

    }

}
```

### 29.4.6.7 Write Disabling to DIMM SPD EEPROM Addresses

Although this controller is not intended to participate in the DIMM Serial Presence Detect (SPD), an attacker connects its data and clock lines to the segment which includes the SPD EEPROM and mounts an attack. To prevent this, a write disable (MCTRL.SPDDIS) is introduced. If SPDDIS is deasserted, writes are not restricted; however, if SPDDIS is set then writes to the address range A0h-AEh are blocked and an error is flagged. Writes to addresses outside the range A0h-AEh are not affected by SPDDIS, and reads are never affected.

**Table 29-16. DIMM SPD EEPROM Write-Disable Mechanism**

| Target Address [7:4] | Target Address [0] (Read/Write Operation) | SPD Disable Bit (MCTRL.SPDDIS) | SMBus Behavior |
|---|---|---|---|
| Ah | 0 (write) | 0 (enabled) | Allow writes to addresses A0-AEh. |
| Ah | 0 (write) | 1 (disabled) | Deny writes to addresses A0-AEh and log error (ERRSTS.SPDWE). |
| Ah | 1 (read) | Any | Allow reads to addresses A0-AEh. |
| != Ah | Any | Any | Allow writes and reads. |

The SPDDIS is read-write-once, and the BIOS Memory Reference Code (MRC) is expected to set this bit when SPD is complete. Since the SMBus (host) controller is not intended for SPD, the BIOS must set its SPDDIS at any time in the boot flow.

## 29.4.7 SMT as Target

The SMT has a fully-functional target interface for other masters on the SMBus intending to communicate with the SMT. The usage models for this are:

- SMBus ARP mastering or ARP target
- Embedded Controller (EC) on the SMBus communicating with the SoC

The hardware aspect of the target interface is highly generalized. Most transactions are treated as raw data which are pushed to the firmware, where the firmware transacts level activity like protocol detection. Acceleration in hardware is limited to interception of ARP Get UDID and SMBus block-read protocols, which require an immediate return of data to an external master and PEC CRC calculation. The other hardware responsibility is inspection of enough transaction bytes (e.g., target address, command code, UDID) to ascertain the protocol type and to engage the appropriate hardware flow.

### 29.4.7.1 Hardware Buffering for Target Support

The hardware implements a separate 240-bytes buffer to store the bytes it receives as a target, but when SMT is target to receive SMBus block write/read, the support maximum payload is <=32B.

The hardware also implements a separate 32-byte buffer which provides generic read data to external masters for any reads that they perform. This is a generic usage model in which firmware repeatedly programs the generic read-data buffer register (GPBRDBUF) with data, programs a MMIO offset (GPBRCTRL) with address, command, and byte count of how much data is present in the generic read data buffer, and communicates to the external master the hardware is ready for taking the read (in-band through the SMBus). The external master then launches a block read to the address programmed with the command, and the hardware provides the byte count and data bytes associated with it.

*Note:* When SMT as slave receives I2C write, the maximum buffer size supported can be 240 bytes with custom I2C driver.

### 29.4.7.2 Target Descriptor

Unlike the master descriptor, the target descriptor is a ring buffer of data and status, all existing concurrently. This simplified model results in the hardware writing a status of a received cycle immediately followed by the data received in the cycle (if applicable).

See also TRxSTS, which is used for debug.

**Figure 29-7. Target Ring Buffer**

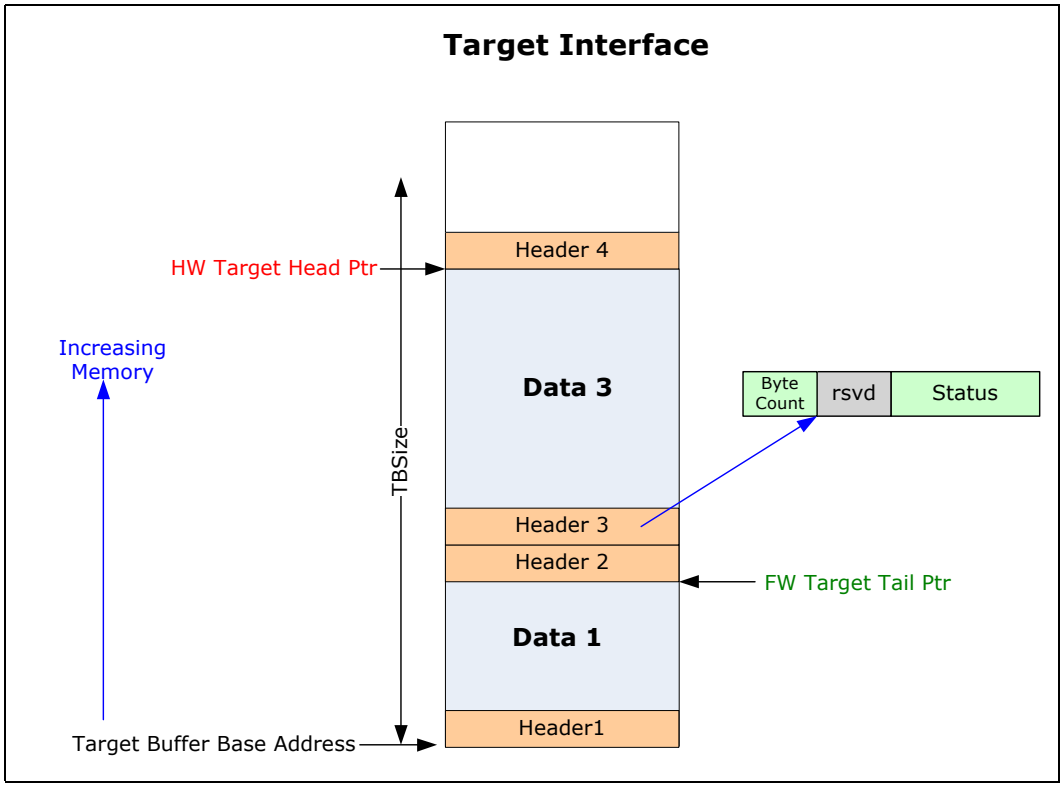
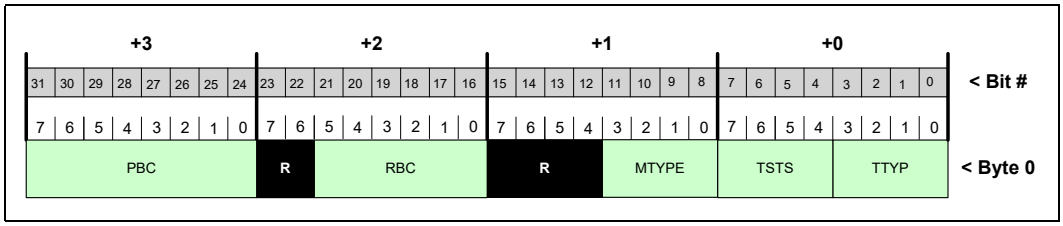



**Figure 29-8. Target Header Format**



## Table 29-17. Target Header Descriptor (Sheet 1 of 2)

| Bit # | Field | Description |
|---|---|---|
| 31:24 | PBC | **Payload Byte Count:** The hardware updates this field to indicate how many bytes of payload data follows after the header in the target buffer. For writes coming in, this data includes the sum total of all bytes ACKed from start to stop. The hardware does then DMA all those bytes to memory.<br>• For error scenarios where the master drives more data than expected or command code does not match, the hardware does DMA all bytes ACKed on the SMBus from start until the first byte NACKed by the hardware (this byte is not sent to memory), and the total number of bytes DMAed to memory is reflected in this field.<br>A value of all zeros in this field means no payload is following the header. |
| 23:22 | Reserved | Reserved |
| 21:16 | RBC | **Read Byte Count**: For external master-generated reads, this data includes the sum total of bytes provided by the hardware which were successfully transferred over SMBus. For example: for an externally generated block read, the hardware provides byte count, X number of bytes, and PEC (if PEC-enabled transaction). In this case, the RBC field is 1+X+1 (for PEC).<br>***Note:*** For SMBus block reads, the last byte transferred is NACKed by the external master per definition. This is a data byte or PEC byte supplied by the hardware.<br>The data bytes are not written to memory, only the byte count indicates how many bytes provided by the hardware were successfully transferred over SMBus.<br>This field is all zeros if the transaction was an externally-generated write.<br>The header is written to memory (if enabled to do so) containing the status of the transaction.<br>For error scenarios in reads, the status register in the header provides details of the error, and this field contains incorrect values due to the pipelined nature of the hardware. |
| 15:12 | Reserved | Reserved |
| 11:8 | MTYPE | **Message Type**: These bits indicate which expected transaction happened on the bus since they are expected in the usage model and some have pre-programmed data provided by the firmware.<br>• 0000: SMBus/$I^2C$ transaction<br>• 0111: Block read to general purpose read data buffer<br>• 1000: SMBus ARP prepare to ARP<br>• 1001: SMBus ARP reset device (general)<br>• 1010: SMBus ARP Get UDID (general)<br>• 1011: SMBus ARP assign address<br>• 1100: SMBus ARP Get UDID (directed)<br>• 1101: SMBus ARP reset device (directed)<br>• 1111: SMBus ARP notify ARP master or SMBus host notify<br>• Others: Reserved<br>***Note:*** Firmware: The cycle does not progress far enough for the hardware to encode the correct encodings. In this case, the hardware inserts some reserved value. |

## Table 29-17. Target Header Descriptor (Sheet 2 of 2)

| Bit # | Field | Description |
|---|---|---|
| 7:4 | TSTS | **Transaction Status**: This field indicates the overall completion status of the transaction, i.e., success, fail, error condition, etc. Multiple error conditions occur in the same transaction (e.g., the hardware NACKs a byte and instead of an external master signaling stop, it continues to toggle the clock and eventually holds the clock low and causes time-out). The hardware captures the first error condition, i.e., the register reads 0011 for the example listed above.<br>Refer to Table 29-19, "Target Header Encodings (TSTS) Per Transaction Type (TTYPE)" on page 770.<br>• 0000: Success with no errors.<br>• 0001: Speculative PEC error detected (the hardware always presumes PEC-enabled writes received).<br>• 0010: Protocol error, i.e., the external master violated the SMBus protocol. For example, this is asserting STOP in the middle of the byte, or a stopped toggling clock, or doing an illegal repeated start, collision, etc.<br>• 0011: Hardware NACK, i.e., the hardware NACKed one or more bytes as the byte did not match the expected byte of the sequence or exceeded hardware limitations.<br>***Note:*** This is the hardware overflow and does not comprehend overflow of an individual protocol (e.g., block write, received bytes > indicated byte count).<br>• 0100: External NACK (external master NACKed at least 1 byte supplied by the hardware on an externally generated read).<br>• 0101: Clock-low time-out<br>• 0110: Data-low time-out<br>• Others: Reserved |
| 3:0 | TTYPE | **Transaction Type**: This field indicates the type of transaction received by the hardware in terms of the various usage models supported:<br>• 0000: Default SMBus transaction to C2h<br>• 0001: Transaction targeting address (TACTRL.ADDR0) of UDID0<br>• 0010: Transaction targeting address (TACTRL.ADDR1) of UDID1<br>• 0100: Host notify transaction targeting 10h<br>• 0101: Generic programmable block read to the programmed address in GPBRCTRL.GPTRADR<br>• Others: Reserved<br>***Note:*** This definition is consistent with TSTS.TCIP. |

Only certain combinations of target descriptor MTYPE and TTYPE are valid. These are summarized in Table 29-18 (shaded cells and all unlisted combinations are invalid). For each valid (MTYPE, TTYPE) pair a reference is included to the detailed hardware flowchart.

### Table 29-18. Valid Target Descriptor MTYPE and TTYPE Combinations

| | | MTYPE | | | |
|---|---|---|---|---|---|
| | | **ARP-Related**<br>**1000**<br>**1001**<br>**1010**<br>**1011**<br>**1100**<br>**1101** | **SMBus/I²C**<br>**0000** | **Notify Host**<br>**1111** | **GPBR**<br>**0111** |
| **TTYPE** | 0000 | Section 29.4.7.5.2 | Invalid | Invalid | Invalid |
| | 0001<br>0010 | Invalid | Section 29.4.7.5.6 | Invalid | Invalid |
| | 0100 | Invalid | Invalid | Section 29.4.7.5.1 | Invalid |
| | 0101 | Invalid | Invalid | Invalid | Section 29.4.7.5.3 |

### 29.4.7.3 Target Transaction Status

Table 29-19 provides a reference of the encodings of the transaction status (TSTS) nibble in the header WB for all valid target cycles (TTYPE).

**Table 29-19. Target Header Encodings (TSTS) Per Transaction Type (TTYPE) (Sheet 1 of 2)**

| TTYPE Cycle Type | TSTS 0000 Success | TSTS 0001 PEC Error | TSTS 0010 Protocol Error | TSTS 0011 Hardware NACK | TSTS 0100 External NACK | TSTS 0101 Clock Low Time Out | TSTS 0110 Data Low Time Out |
|---|---|---|---|---|---|---|---|
| 0101<br>GP Block Read | No errors | N/A (The hardware transmits the PEC.) | • A collision is detected by the hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus.<br>• The external master signals stop/restart, and so on in the middle of a byte.<br>• The external master signals a stop on a byte boundary before the byte count expired.<br>• The external master continues driving more clocks even after the hardware has provided all the bytes.<br>• The external master drives ACK instead of NACK on the PEC byte sent by the hardware.<br>• Bits [7:1] of the repeated start address do not match the GPBR address.<br>• Bit 0 of the repeated start address is 0, and the policy is to check for 1. | The command code byte received from the external master does not match the expected value. | The external master NACKs at least one of the bytes provided by the hardware, i.e., Byte Count, Data1, Data2, Data3,..., DataN. | The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing. | The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing. |

## Table 29-19. Target Header Encodings (TSTS) Per Transaction Type (TTYPE) (Sheet 2 of 2)

| TTYPE Cycle Type | TSTS 0000 Success | TSTS 0001 PEC Error | TSTS 0010 Protocol Error | TSTS 0011 Hardware NACK | TSTS 0100 External NACK | TSTS 0101 Clock Low Time Out | TSTS 0110 Data Low Time Out |
|---|---|---|---|---|---|---|---|
| 0001<br>Cycles to ADDR0/UDID0<br>or<br>0010<br>Cycles to ADDR1/UDID1<br>or<br>0000<br>Cycles to Default Address | No errors | PEC received from the external master did not match the hardware-calculated PEC.<br>***Note:*** This is a speculative error only since the firmware confirms it was a PEC-enabled SMBus transaction (see Section 29.4.7.5.6). | • A collision is detected by the hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus.<br>• The external master signals stop/restart, and so on in the middle of a byte.<br>• The external master signals stop on a byte boundary before the byte count expired.<br>• The external master drives more bytes than indicated by the byte count.<br>• The external master generates a start instead of a stop on a byte boundary as a response to the hardware NACK in the previous data phase. | • The hardware NACKs when the external master drives more data than the limit programmed in TRxCTRL.MRxB<br>• The hardware detects the protocol/address violation (see Table 29-8 and Table 29-9). | N/A (The hardware does not provide any bytes.) | The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing. | The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing. |
| 0100<br>Host Notify *or* Notify ARP Master | No errors | N/A (no PEC) | • A collision detected by the hardware. This happens when the hardware intended to drive NACK and saw an ACK on the bus.<br>• The external master signals stop/restart, and so on in the middle of a byte.<br>• The external master signals a stop on a byte boundary before it sends three data bytes.<br>• The external master drives more than three data bytes after the address phase. | N/A (The hardware does not check any bytes.) | N/A (The hardware does not provide any bytes.) | The hardware detects the SMBus clock low time-out anywhere in the middle of the transaction that it is actively servicing. | The hardware detects the SMBus data low time-out anywhere in the middle of the transaction that it is actively servicing. |

*Note:* All references to the TRxCTRL.MRxb registers must take into account the description of MRxB, i.e., the value is not directly matched since it takes into account some arithmetic based on $I^2C$/MCTP protocol.

*Note:* For time-out detection as a target, the hardware must be actively servicing the transaction when the time-out occurs. This means the hardware must have at least ACKed the address phase on the SMBus and must be committed to the transaction based on the settings of the policies in SUSCHKB register.

### 29.4.7.4 Target Memory Buffer Hardware-Firmware Flow

As described in the target descriptor section, a ring buffer is maintained by the firmware where the hardware sends the notifications to the firmware. The various notifications are:

1. External master initiates a read: notification of a success/failure is sent to the firmware in header. Read data is returned by the hardware (data is pre-programmed by the firmware).
2. External master initiates a write: notification of a success/failure is sent to the firmware in the header. Write data received by the hardware is sent as a payload to the memory.

*Note:*

1. The target buffer is empty when HWtHeadPtr = FWtTailPtr (HTHP = FTTP).
2. The target buffer is full when (HTHP = FTTP - 4B) or (HTHP - FTTP = TBS).
3. A buffer wrap for HTHP is when HTHP = TBS. When this exists, the hardware writes Dword to memory and then rolls over to 0x0000 unless a buffer full condition exists, i.e., FTTP = 0x0000.
4. The firmware must never increment the FWtTailPtr to a value greater than HWtHeadPtr.
5. If the target ring buffer is N-bytes deep, only N-1 bytes are utilized since the hardware/firmware does not implement a wrap bit.

#### 29.4.7.4.1 Initialization

1. The firmware allocates a buffer in the firmware memory to be used as the target ring buffer as one contiguous space.
2. The firmware then programs up the Target Buffer Base Address (TBBA) register with a 64B-aligned memory address of the ring buffer.
3. The firmware assigns the Target Buffer Size (TBS) register with the actual size of the ring buffer with the maximum limit of 64 KB.
4. The hardware reset initializes HWtHeadPtr and FWtTailPtr to 0.
5. The firmware enables all target addresses as needed, and programs the register-based read data as needed.
6. The firmware also programs interrupts as needed.
7. Finally, the firmware sets the TPOLICY.TGTEN bit to enable the target logic.

#### 29.4.7.4.2 Hardware-Firmware Flow

Once the firmware has completed initialization, the hardware target logic continues to idle until an external SMBus master sends a transaction.

**External Master Initiating Writes:**

1. Once the hardware receives a write transaction targeting one of its slave addresses, the hardware responds to the address and command phases.
2. Each byte ACKed is eventually DMAed to the memory starting at location Base + HWtHeadPtr + 4B as Dword writes.
3. Once the transaction is completed, the hardware writes the final bytes to memory (PBC field in the header indicates to the firmware how many true bytes are in the payload).
4. Upon completion of the transaction, the hardware writes the header Dword of that transaction into the location pointed to by HWtHeadPtr. This contains all the status information for the transaction. (See Figure 29-9, “High-Level Target Flow” on page 775.)
5. The hardware then updates the HWtHeadPtr to the next free Dword location in the target memory.
6. The hardware then writes interrupt information to the Dwords as pointed to by SMTICL. (See Table 29-20, “Target Transaction Behavior Due to SUSCHKB.IRWST” on page 776.)
   a. The hardware writes the current value of the HWtHeadPtr to TRGT.HTHP and sets TRGT.VALID.
   b. Also, if the transaction terminated abnormally the error condition is written (e.g., to ERR.TRBAF, ERR.TRBF, ERR.CKLTO) and ERR.VALID is set to indicate an error was present.
   c. Finally, if enabled, the hardware then sends MSI to the firmware.

**External Master Initiating Reads:**

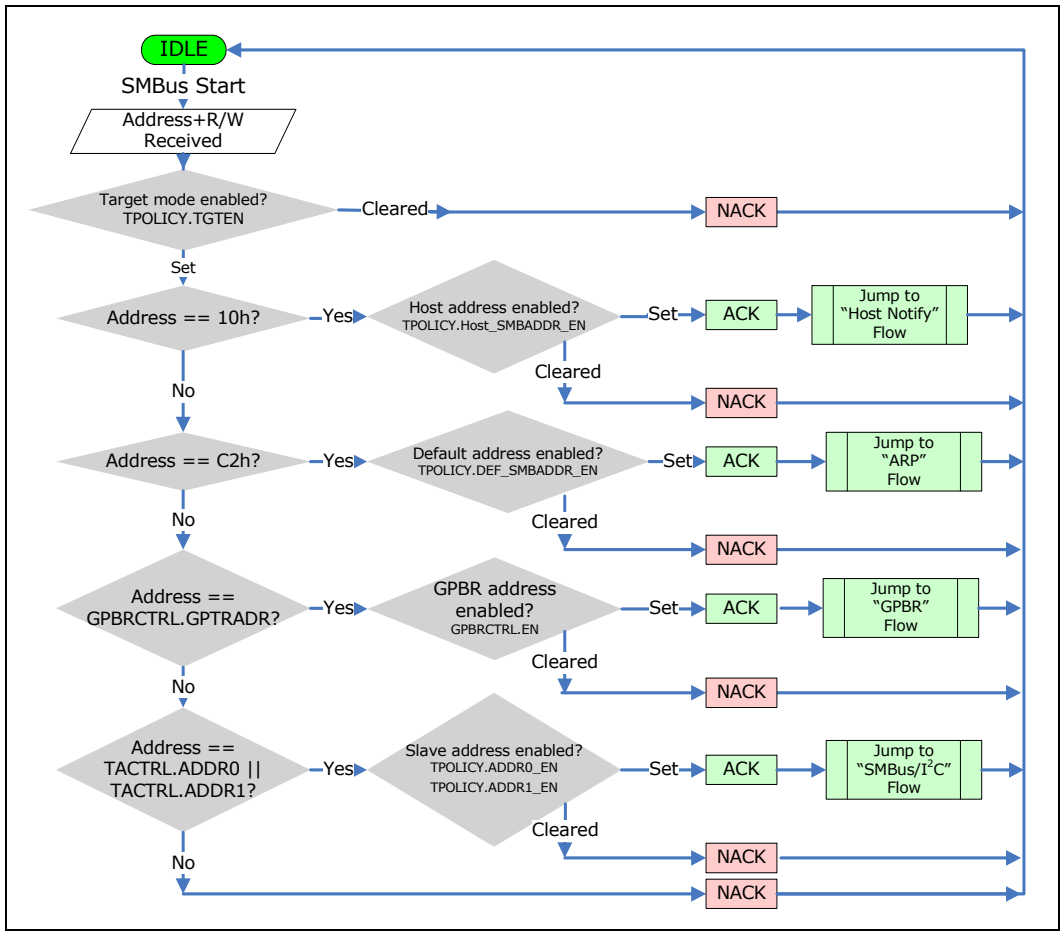
1. Once the hardware receives a read transaction targeting one of its slave addresses, the hardware checks the address, command, and any other bytes sent by the external master and ACK/NACK appropriately.
2. If all the bytes sent by the external initiator are ACKed by the hardware, the hardware provides read data to the initiator. (See Figure 29-9, "High-Level Target Flow" on page 775.)
   a. If the initiator takes all the bytes as expected and terminates the transaction normally, depending on the state of TCTRL.SCHWBP bit, the hardware creates (or does not create) a header Dword and writes to memory at a location pointed by HWtHeadPtr with the BC field updated.
   b. If error conditions are read, the hardware updates the error status registers in the status Dword, creates the header with the appropriate BC field, and writes to memory at a location pointed by HWtHeadPtr.
3. The hardware then updates the HWtHeadPtr to the next free Dword location in the target memory.
4. The hardware then writes interrupt information to the Dwords as pointed to by SMTICL. (See Table 29-20, "Target Transaction Behavior Due to SUSCHKB.IRWST" on page 776.)
   a. The hardware writes the current value of HWtHeadPtr to TRGT.HTHP and sets TRGT.VALID.
   b. Also, if the transaction terminated abnormally, the error condition is written (e.g., to ERR.TRBAF, ERR.TRBF, ERR.CKLTO), and ERR.VALID is set to indicate an error was present.
   c. Finally, if enabled, the hardware then sends an MSI to the firmware.

### 29.4.7.5 Target Flow

Target flow for the hardware depends on the target address seen on the SMBus. The hardware has deterministic behavior based on the address. A high-level flowchart is shown in Figure 29-9. It does not capture all possibilities of errors, but aims to highlight the sequence of events. Depending on the received address, detailed flowcharts follow in later sections for four protocol categories.

The hardware address matching after an SMBus start condition is subject to SUSCHKB.IRWST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart. It does have implications for how the Quick command, receive byte, and $I^2C$ read are handled.

**Figure 29-9. High-Level Target Flow**



### Table 29-20. Target Transaction Behavior Due to SUSCHKB.IRWST

| | Quick Command | | Receive Byte | $I^2C$ Read |
|---|---|---|---|---|
| **IRWST** | **R/W# Bit = Read** | **R/W# Bit = Write#** | **R/W# Bit = Read** | **R/W# Bit = Read** |
| 0 | Transaction is ACKed and then dropped. No descriptor is sent.<br>The hardware incorrectly invalidates a supported command option. | Transaction is ACKed and a descriptor is sent to the ring buffer for the firmware. | Transaction is ACKed and then dropped. No descriptor is sent.<br>The hardware invalidates this unsupported command. | Transaction is ACKed and then dropped. No descriptor is sent.<br>The hardware invalidates this unsupported command. |
| 1 | Transaction is ACKed and a descriptor is sent to the ring buffer for the firmware. | Transaction is ACKed and a descriptor is sent to the ring buffer for the firmware. | Transaction is ACKed and a descriptor is sent to the ring buffer for the firmware.<br>The command is aliased and treated by the hardware as a send byte.<br>The firmware must invalidate (drop) this unsupported command by detecting the R/W# bit is set in the descriptor. | Transaction is ACKed and a descriptor is sent to the ring buffer for the firmware.<br>The command is aliased and treated by the hardware as an $I^2C$ write.<br>The firmware must invalidate (drop) this unsupported command by detecting the R/W# bit is set in the descriptor. |

*Note:* Based on the interaction between SUSCHKB.IRWST and certain supported and unsupported commands, the following recommendations are made:

- If a Quick command is unused (or if its usage is restricted to R/W# bit cleared), then IRWST is cleared. The hardware invalidates the unsupported commands receive byte and $I^2C$ read.
- Otherwise, IRWST is set, and the firmware invalidates (drops) both of these unsupported commands.

#### 29.4.7.5.1 Host Notify Target Flow

**TTYPE = 0100** and **MTYPE = 1111**

The host notify flow comprehends both the ARP host notification and the ordinary SMBus host notify protocols since they differ only in the content of the second received byte. If the firmware has declared this address to be busy (TPOLICY.HOSTBSY), then the hardware will ACK the address byte but NACK all subsequent bytes.

#### 29.4.7.5.2 SMBus ARP Target Flow

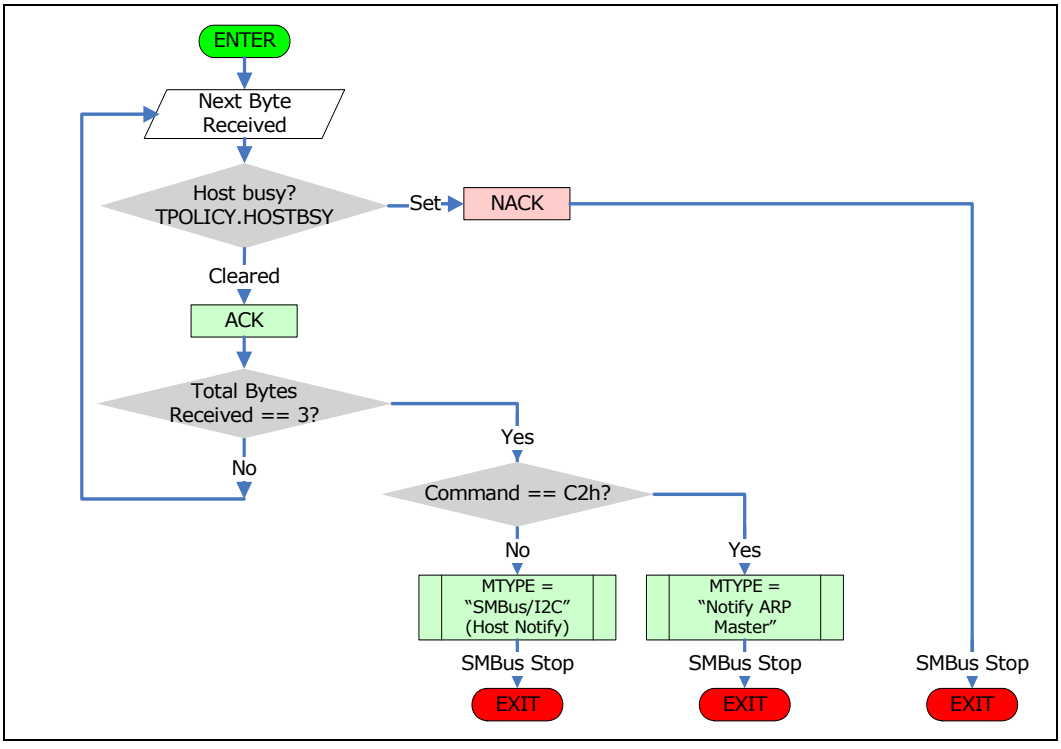
**TTYPE = 0000** and **MTYPE = 1000, 1001, 1010, 1011, 1100, 1101**

The ARP target flow comprehends all ARP protocols, including general and directed flavors. The priority shown in the flowchart for decoding the command byte is illustrative only. All ARP-related transactions require PEC processing: for Get UDID, the hardware must calculate and transmit the PEC. In all other protocols the hardware must calculate the PEC and compare that against the received PEC byte.

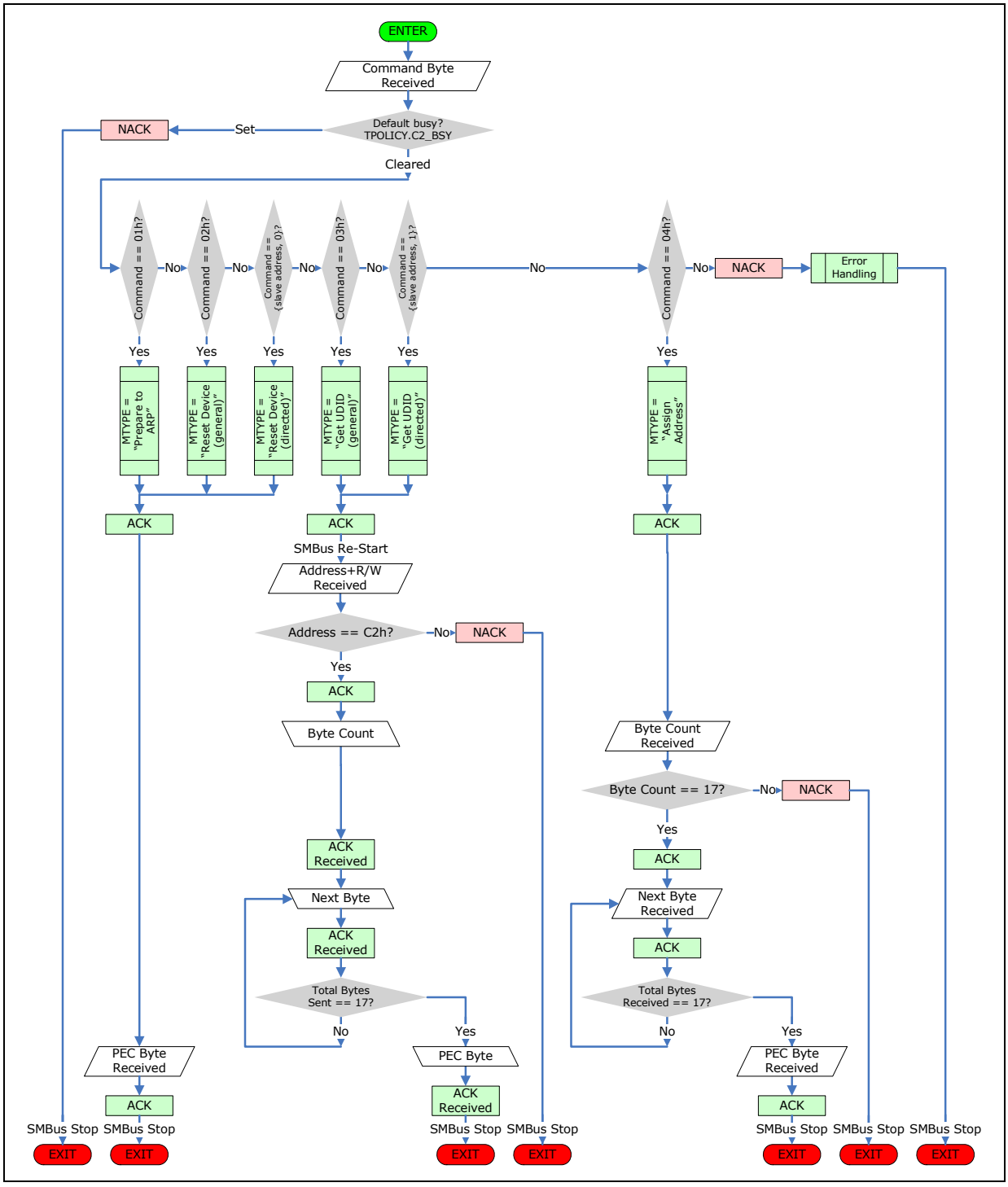
If the firmware has declared this address to be busy (TPOLICY.C2_BSY), then the hardware will ACK the address byte but NACK all subsequent bytes.

The hardware address matching after an SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

**Figure 29-10.Host Notify Target Flow**



## Figure 29-11.SMBus ARP Target Flow



#### 29.4.7.5.3 General-Purpose Block Read Flow

**TTYPE = 0101** and **MTYPE = 0111**

SMT provides a mechanism for an external master to read up to 32B of data from the SMT.

The rules are:

- The external master cannot asynchronously launch a read request.
- The address to read from is programmed by the firmware (GPBRCTRL.GPTRADR), and this address must be unique from all other target addresses that SMT supports.
- The external master must read data only as defined by an SMBus 2.0 Block Read command.
- The firmware programs the data into the Data Buffer register (GPBRDBUF) and enables the hardware to support a Read command from the external master (GPBRCTRL.EN).
- The firmware sends a message to the external master requesting it to read the data.

The sequence of events is indicated by the flowchart shown in Figure 29-12.

The hardware matches the received command code against a Firmware-programmed Expected command (GPBRCTRL.CMD). A mismatch causes the transaction to be terminated with a NACK. This matching is required since GPBR is handled entirely in the hardware; the firmware does not participate in validating the address and command.

An appended PEC is optional and is enabled by the firmware (GPBRCTRL.PECEN). Its usage is determined by high-level policy negotiation between the SoC and the external master.

Nominally, the hardware clears its enable bit immediately after ACKing the command byte. This enables a handshake between the hardware and the firmware such that the hardware transmits read data only after being programmed and its trigger set by the firmware. Alternately, the hardware is enabled (GPBRCTRL.HWCLRDIS) to continually respond to read requests from an external master; however, in this mode, determinism is not guaranteed between the firmware programming to and the master reads of the data registers.

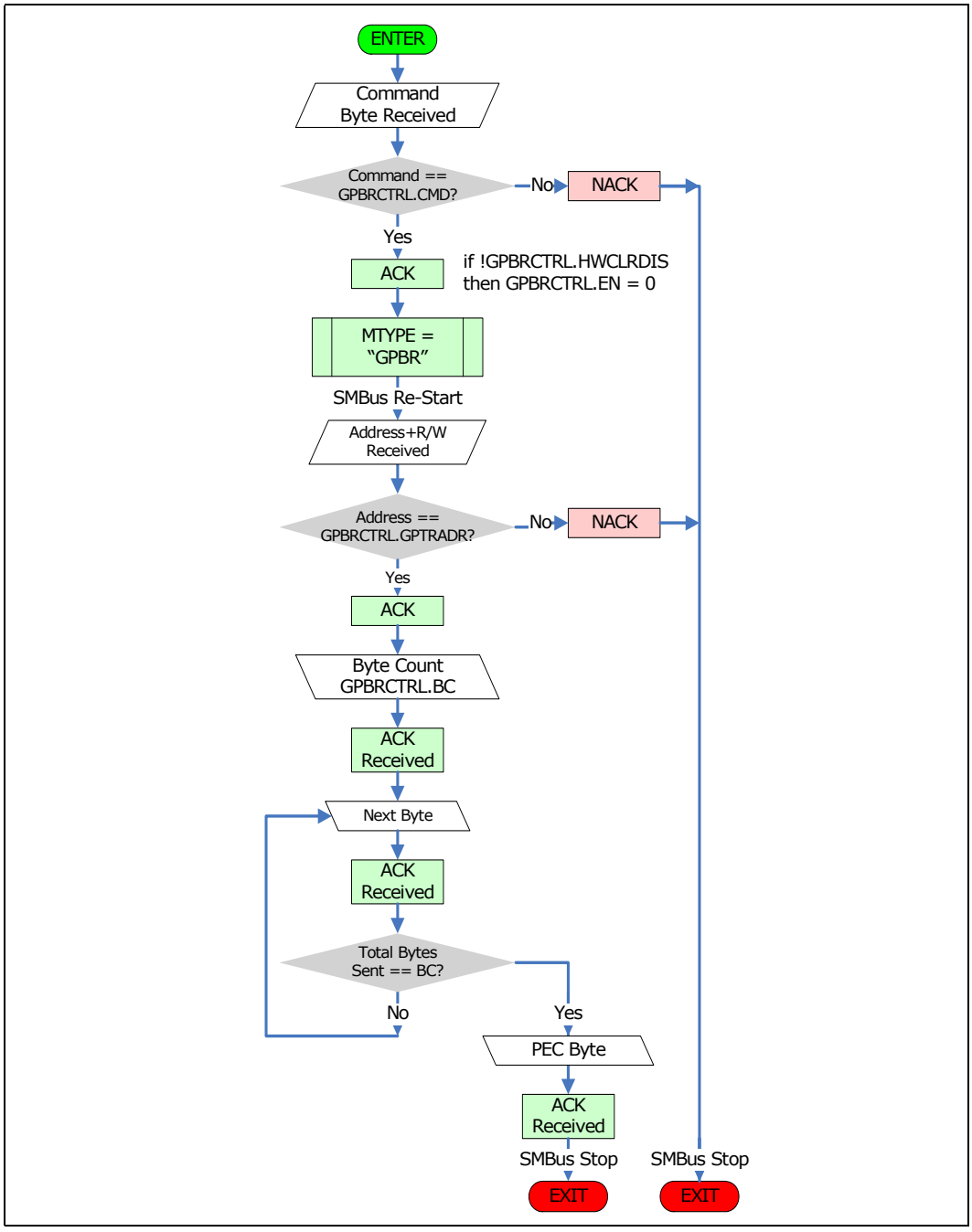
The hardware address matching after an SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

The generic programmable read data buffer register (GPBRDBUF) loads data into the generic programmable read data buffer. The buffer provides a mechanism of supporting Generic Programmable Reads (GPBR) by an external SMBus master under the firmware initiated flow.

The data buffer is filled by the firmware initiating DW writes to this register. A hidden pointer always points to the next available buffer location—reading and writing to GPBRDBUF cause the pointer to increment automatically.

*Note:* Because the pointer auto-increments after each read and write, a read of GPBRDBUF does NOT return the most recently written data. Normal usage requires the buffer contents to be cleared before filling it with a new data payload. Therefore, a read (of the next empty location) returns 0s. (See also GPBRCTRL.BUFRST and GPBRCTRL.PTRRST.)

**Figure 29-12.General Purpose Block Read with PEC Target Flow**



#### 29.4.7.5.4 Normal Usage

Before programming new data into the GPBR buffer, the firmware sets both GPBRCTRL.BUFRST (which resets the buffer contents to 0) and GPBRCTRL.PTRRST (which resets the buffer pointer position). After the hardware has completed the reset, it clears both bits.

#### 29.4.7.5.5 Debug Usage

To enable the firmware to read back (verify) the contents of the GPBR buffer, the firmware sets only GPBRCTRL.PTRRST, which resets the buffer pointer position but leaves the buffer contents intact. Consecutive reads then return the previously written buffer data.

The hardware provides read data payload starting from bits [7:0] of Dword0 of the buffer, then bits [15:8] of Dword0, and so on. All bytes must be programmed in the sequence to be sent out without gaps. The hardware counts the number of bytes indicated in the GPBRCTRL.BC register and starts sending them out sequentially starting at byte 0 of Dword 0 until the byte count is decremented to 0.

#### 29.4.7.5.6 SMBus/I$^2$C Target Flow

**TTYPE = 0001, 0010** and **MTYPE = 0000**

SMT supports being a target of certain SMBus and I$^2$C transactions (see Figure 29-13, "SMBus/I$^2$C Target Flow" on page 782), and it acknowledges transactions directed to either of its two target addresses (TACTRL.ADDR0 or TACTRL.ADDR1). The basic flow is depicted in Figure 29-13. In general the target hardware interface is agnostic of command and PEC bytes. The firmware handles these bytes.

*Note:* It is assumed that block reads to SMT as a target are directed only to the dedicated GPBR address (GPBRCTRL.GPTRADR) and are not considered in this flow.

Write cycles received by the SMT target interface nominally terminate when the external master produces a stop condition. A premature hardware NACK occurs if the interface is busy or overflows. The SMBus Quick command is comprised only of the address byte, with the data encoded entirely within the R/W# bit. All other SMBus and I$^2$C transactions have additional data bytes.

The optional PEC byte is not shown explicitly because the firmware determines the actual presence of PEC. This provides for PEC to be enabled on a per-function, per-address, per-protocol, or even per-transaction basis. The hardware always performs a **speculative** internal PEC calculation, which it then compares to the final received byte. If the received and calculated bytes match, this is a strong indication that the transaction was PEC-enabled, but the firmware has the final authority. Conversely, a mismatch is not proof of a transmission error—it may have been a PEC-less transaction. In this case, the firmware ignores the PEC status flag in the target descriptor TSTS field, which resulted from the hardware invalid speculative of the PEC comparison.

Since some protocols have identical transmission templates it is also the responsibility of the firmware to prevent or resolve apparent aliasing (e.g., aliasing of a send byte with the PEC and a write byte; aliasing of a write byte with the PEC and a write word; aliasing of the SMBus write byte and I$^2$C MTx-to-SRx). It is assumed this is accomplished through negotiation between the firmware and the external master. For instance, the command byte differentiates protocols or certain protocol combinations are excluded.

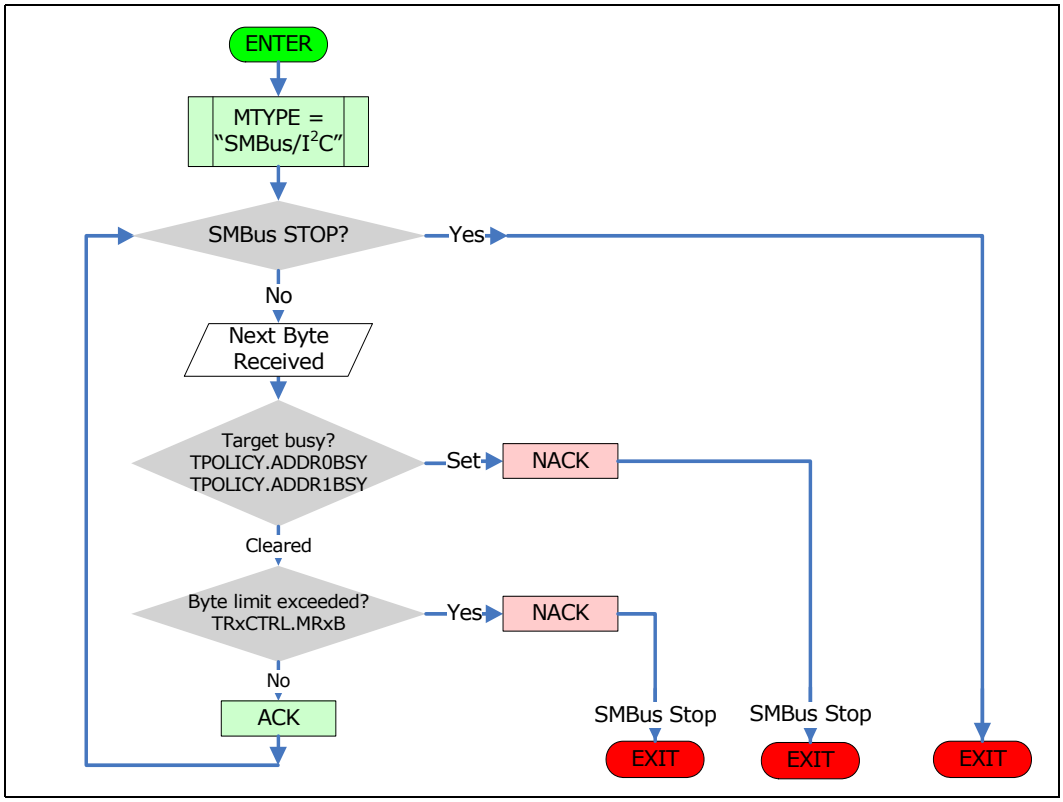
A special case of aliasing occurs for the SMBus block write. Since the hardware ignores the command byte, it cannot distinguish the block write byte count from an ordinary data byte. Therefore, the hardware captures and pushes all bytes to the firmware for processing, and the hardware is not required to verify the received byte count. In the target mode, the firmware programs a ceiling (TRxCTRL.MRxB) on the write length such that if a transaction overflows the ceiling, then the hardware must NACK any unexpected data bytes. This ceiling must not exceed the buffer size. The hardware NACK flag in the target descriptor TSTS field is set upon a hardware (buffer) overflow, but this flag does not comprehend overflow of an individual protocol (e.g., a block write where received bytes > the indicated byte count).

*Note:* If a target address receives both $I^2C$ and SMBus protocols, the maximum permitted write length (TRxCTRL.MRxB) must accommodate both. For example, if a target receives an SMBus block write with PEC, the write length must be greater than or equal to 36 bytes (= address + command + byte count + N data bytes + PEC, where N<=32).

If the firmware has declared a matched address to be busy (TPOLICY.ADDR0BSY or TPOLICY.ADDR1BSY), then the hardware will ACK the address byte but NACK all subsequent bytes.

The hardware address matching after any SMBus repeated-start condition is subject to SUSCHKB.IRWRST; if asserted it causes the hardware to ignore the R/W# bit during matching. This behavior is not shown in the flowchart.

**Figure 29-13.SMBus/$I^2C$ Target Flow**



## 29.4.8 Dynamic SMT Policy Update

The hardware provides a mechanism for the firmware to change the policies of the hardware and other registers by the firmware without requiring a reset to the system.

### 29.4.8.1 Master Policy

For updates to the master side of the registers, the firmware already has the control by stopping the hardware DMA engine and halting new master transactions. This ensures that the hardware master side is completely disabled and the firmware chooses to update the policy and registers.

### 29.4.8.2 Target Policy

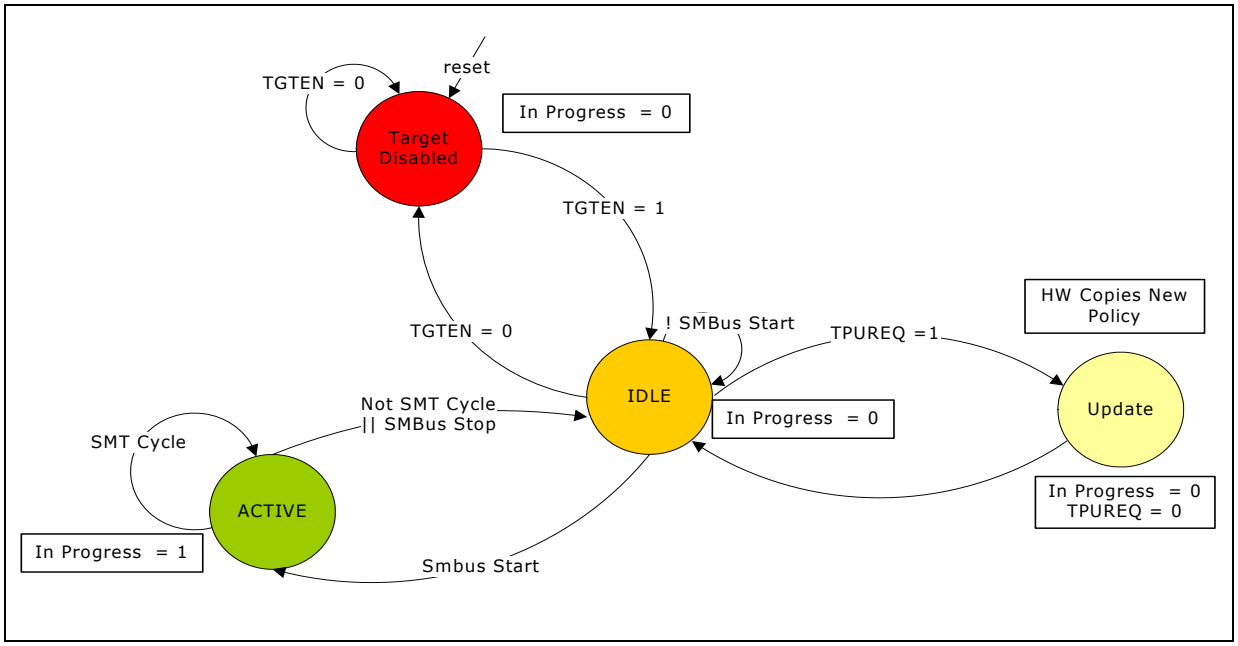
Updates to the target policy require a flow, since the hardware asynchronously receives a target cycle.

The hardware maintains a current target policy internally without direct visibility to the firmware. The firmware sees the current policy by reading the TPOLICY register. When TPOLICY.PTUREQ=0, it means the current policy settings are effective in the hardware (assuming the firmware follows flow and did not write to any other register without first setting the PTUREQ bit). To change the policy, the firmware first disables the target completely or selectively disables addresses so while the firmware is extensively updating registers, the hardware NACKs appropriately in the address or command phase as desired. After that is done, the firmware must again update the policy following the same flow to have the final desired policy in effect.

1. The firmware sets the intermediate policy desired by programming the TPOLICY register with TPOLICY.PTUREQ also set.
2. When the hardware reaches the SMBus idle state, it checks if PTUREQ is:
   a. If set, the hardware captures the current setting of the policy in PTUREQ and loads into its internal registers. It also clears the PTUREQ bit once the policy is captured.
   b. If clear, the hardware continues to be in idle until another SMBus start is seen.
3. The hardware continues to act on an intermediate policy resulting in NACK on the address phase or on the command phase of incoming cycles (programmed by the firmware).
4. The firmware polls or reads the PTUREQ register to make sure it is clear before reprogramming the registers, etc.
5. Once the firmware is ready to ungate the hardware, the firmware again follows step #1 in which it sets the new policy.
6. The flow again follows steps 2-4 before the final firmware policy is set and active in the hardware.

A conceptual flow is shown in Figure 29-14.

**Figure 29-14.Target Dynamic Policy Update**



## 29.5 Interrupts

The SMT hardware has the following causes of interrupts (if enabled).

1. Master Interrupt - The hardware generates master interrupts for transactions ending successfully or in a failure.
2. Target Interrupt - Generated whenever the hardware writes data and/or the header to memory.
3. Error Interrupt - Generated whenever the hardware detects a bus error or a resources error.

**Table 29-21. Summary of SMT Interrupt Enables and Sources**

| Interrupt Enable/Mask | Interrupt Source | Governed By | Asserted Upon |
|---|---|---|---|
| **Master Interrupts** | | | |
| Master Descriptor INT field | MSTS.MIS | | Successful completion of master cycles |
| MCTRL.MEIE | MSTS.MEIS | | Unsuccessful completion of master cycles |
| **Target Interrupts** | | | |
| TCTRL.TIE | TSTS.TIS | TCTRL.SCHWBP<br>TCTRL.UCHWBP<br>TCTRL.URxTWP | Successful completion of target cycles<br>Unsuccessful completion of target cycles<br>Unsuccessful completion of write to target |
| **Error Interrupts** | | | |
| ERRINTMSK.CKLTO | ERRSTS.CKLTO | | SMBus clock-low time-out |
| ERRINTMSK.TRBF | ERRSTS.TRBF | | Target ring buffer is full. |
| ERRINTMSK.TRBAF | ERRSTS.TRBAF | | Target ring buffer is almost full. |
| ERRINTMSK.IHIE | ERRSTS.IHIE | | Read completion with non-successful status |
| ERRINTMSK.IMAE | ERRSTS.IMAE | | Request is made when PCICMD.BME is clear. |
| ERRINTMSK.ITE | ERRSTS.ITE | | Error in IPI transmit transaction |
| ERRINTMSK.IRDPE | ERRSTS.IRDPE | | Data parity error in IPI receive transaction |
| ERRINTMSK.IRE | ERRSTS.IRE | | Error in IPI receive transaction |
| ERRINTMSK.SPDWE | ERRSTS.SPDWE | MCTRL.SPDDIS | Write is attempted to SPD address range |
| ERRINTMSK.CPE | ERRSTS.CPE | | CSR parity error |

*Note:* First-error cause information is logged in ERRINFO. Also, error interrupts also support masking of escalation to Advanced Error Reporting (AER); see ERRAERMSK.

*Note:* ERRINTMSK.IMAE: this bit is assumed to be 1 when MSICTL.MSIE is set. IMAE occurs only if PCICMD.BME is clear, and MSI requires BME to be set.

### 29.5.1 Master Interrupts

Two causes for a master interrupt are:

1. **Successful** - The Interrupt bit in the descriptor is set to indicate the hardware must generate an interrupt on successful completion of the descriptor. An Interrupt Status Register bit is defined in the MMIO space (MSTS.MIS), which is set every time the hardware retires a descriptor. The enable for this interrupt is defined within each individual descriptor.
2. **Failure** - The descriptor is not successfully completed due to an SMBus error. This interrupt has a Cause Enable (MCTRL.MEIE) and a Cause Status bit (MSTS.MEIS) in the device MMIO space.

Upon completion of descriptor processing, either MSTS.MIS (successful completion) or MSTS.MEIS (unsuccessful completion) must be set by the hardware. This distinction is required so that if the firmware disables MSI (i.e., chooses to operate in the polling mode) by polling both flags, it determines when the descriptor has completed and its completion status.

The master interrupts are sent for the descriptor-based transactions. Therefore, they are serialized and ordered with respect to the descriptor writeback and specifically tied to the descriptor just processed due to the operand requirement.

1. Run the descriptor-based master transaction on the SMBus.
2. Perform the descriptor status WB to memory. (See the status WB field in Figure 29-4, “Master Descriptor Ring Buffer” on page 750, Figure 29-5, “Master Descriptor Format” on page 750, and Table 29-13, “Master Descriptor Field Descriptions” on page 751.)
3. Write 1 to the appropriate Cause Status bit (MCTRL.MEIS or MSTS.MIS).
4. If that appropriate interrupt is enabled (locally and globally), an MSI is sent and the cause status is auto-cleared, else an MSI is not sent and the cause remains set and the firmware is expected to poll the response.

*Note:* MSTS.MEIS itself merely indicates an unsuccessful completion. The descriptor status WB field contains full details of transaction and error conditions.

*Note:* Firmware implementation:

1. If the cause is set for a previous descriptor-based transaction and the firmware enables the global (and local) interrupt enable, the hardware does not send an interrupt for a previously set cause.
2. The firmware is responsible to ensure that if the cause is set, the previous transactions are accounted for before enabling the MSI (globally and locally).

## 29.5.2 Target Interrupts

Since the target ring buffer is a single buffer in the memory, the interrupts sent on behalf of the target are simplified to be sent every time the hardware writes data and/or the header to the memory (depending on if target interrupts are enabled and if global MSIen is set).

The conditions which result in a status header being written are further governed by these policies:

- TCTRL.SCHWBP: upon successful completion of target cycles
- TCTRL.UCHWBP: upon unsuccessful completion of target cycles
- TCTRL.URxTWP: upon unsuccessful completion of writes to the target

This simplified model is used since the hardware supports multiple logical devices and transactions of each type are pushed to memory by the hardware in a sequential manner. The firmware then has to parse each header to determine what each transaction is before the firmware takes the appropriate action.

A single Target Interrupt Enable bit is maintained in the MMIO space (TCTRL.TIE) which gates the sending of this interrupt. An Associated Status bit (TSTS.TIS) is also maintained in the hardware.

Similar to master interrupts, target interrupts are also serialized and ordered. The decision to send an MSI is made after the hardware has written back the header for the received transaction to the memory ring buffer.

1. Receive the target cycle from the SMBus.
2. Perform data and header WB to memory.
3. Write 1 to the appropriate Cause Status bit (TSTS.TIS).

If TCTRL.TIE is enabled (locally and globally), an MSI is sent and the cause status is auto cleared else an MSI is not sent and the cause remains set.

*Note:* Firmware implementation:

1. If the cause is set for a previous target transaction written to memory and the firmware enables the global (and local) interrupt enable, the hardware does not send an interrupt for a previously set cause.
2. The firmware is responsible to ensure that if the cause is set, the previous transactions are accounted for before enabling an MSI (globally and locally).

### 29.5.3 Error Interrupts

The hardware tracks the following conditions for errors:

1. SMBus clock-low time-out: status is set when the hardware observes the SMBus clock is asserted low for more than the value programmed by the firmware.
2. SMBus data-low time-out: status is set when the hardware observes the SMBus data is asserted low for more than the value programmed by the firmware.
3. Target ring buffer almost full: status is set when the hardware detects the target ring buffer has less than 85B free space remaining in the target ring buffer. This check is done after the hardware performs a memory write for data/header.
4. Target ring buffer full: status is set when the hardware in unable to evict its internal buffer/header WB to memory due to lack of space in the target ring buffer. This check is done every time the hardware needs to write to memory.

**Table 29-22. Error MSI Scheduling**

| Event | Global MSIen | Cause Interrupt Enable[1] | Cause Interrupt Status[2] | MSI Action |
|---|---|---|---|---|
| Cause occurs, but interrupts are not enabled. | 0 | 0 | 0 -> 1 | No MSI sent. |
| MSI enable is set, cause enable and cause status are previously set. | 0 -> 1 | 1 | 1 | Send MSI. |
| MSI enable and cause status previously set, cause interrupt enable is set. | 1 | 0 -> 1 | 1 | Send MSI. |
| Cause Status register gets set, interrupts are enabled. | 1 | 1 | 0 -> 1 | Send MSI. |
| Cause Status register previously set, interrupts are enabled, new cause occurs. | 1 | 1 | 1 -> 1 | Does not occur if interrupts are enabled. Cause Status is cleared. If the clearing happens on the same clock as the new cause is set, it is the rule in the next row before. |
| New cause occurs in the same clock as the previous MSI scheduled was sent. | 1 | 1 | 1 -> 1 | Send MSI. |
| Scheduled MSI was sent. | 1 | 1 | 1 -> 0 | The hardware auto-clears the cause register. |
| MSI is scheduled to be sent when MSI enable is cleared. | 1 -> 0 | 1 | 0 -> 1<br>or<br>1 -> 1 | The hardware schedules the MSI or drops it.<br>Any pending MSIs must be reflected in the MSI Pending register. |

1. Cause interrupt mask: see ERRINTMSK and ERRAERMSK.
2. Cause interrupt status: see ERRSTS.

### 29.5.4 Interrupt Cause Logging

Before sending any MSI, the hardware writes information pertinent to the interrupt cause to a memory location identified by Table 29-23.

As indicated in Table 29-23, the Dwords of SMTICL are divided into three categories. The hardware updates only the relevant Dword depending on the nature of the interrupt: interrupts generated during a master transaction involves updates only to Dword #0, likewise target transactions involve only Dword #1. However, if an error condition is present, the hardware requires also writing to Dword #2 to capture the nature of the error.

The firmware always reads all three Dwords, and upon reading them clears any Dword which has non-zero data. (At a minimum two Dwords must be read—master + error or target + error.)

**Table 29-23. Interrupt Cause Information**

| Dword | Bit | Field | Description |
|---|---|---|---|
| **Information from MASTER** | | | |
| 0 | 31 | MSTR.VALID | Master interrupt cause is valid.<br>1: Denotes master status (MSTS), including the master hardware tail pointer, has been written. |
| 0 | 30:0 | Reserved | |
| **Information from TARGET** | | | |
| 1 | 31 | TRGT.VALID | Target interrupt cause is valid.<br>1: Denotes the Target Hardware Head Pointer (HTHP) has been written. |
| 1 | 30:0 | Reserved | |
| **Error Information** | | | |
| 2 | 31 | ERR.VALID | Error interrupt cause is valid.<br>1: Denotes errors status (ERRSTS) has been written. |
| 2 | 30:0 | Reserved | |

# 29.6 SMT RAS Architecture

## 29.6.1 Soft Reset (DEVCTL.IFLR and GCTRL.SRST)

The SMT supports several types of soft reset including:

- Function-level reset as defined by the *PCI Local Bus Specification*, Revision 3.0 (see DEVCTL.IFLR).
- Soft reset directed to the SMT controller (see GCTRL.SRST).

In each case, the soft reset applies only to the specific SMT function addressed. Asserting a soft reset has these effects:

- An immediate and abrupt reset of the SMT master and target logic (which causes a protocol violation of any pending master or target cycles),
- Clear all the MMIO registers except those register bits denoted as sticky or read-write-once (RW-O) or primary-reset-only (PRST) as listed in Table 29-24, and
- Release the SMBus clock and data lines.

**Table 29-24. SMT Soft Reset Exceptions**

| Register Fields | Comments |
|---|---|
| **Sticky (S)** | |
| AERCAPCTL.FEP | |
| AERHDRLOG.TLPHDRLOG | |
| ERRCORMSK.x, x = CIEM, ANFEM | |
| ERRCORSTS.x, x = CIE, ANFE | |
| ERRUNCMSK.x, x = UIEM, UREM, MTLPEM, UCEM, CAEM, CTEM, PTLPEM | |
| ERRUNCSEV.x, x = UIES, URES, MTLPES, UCES, CAES, CTES, PTLPES | |
| ERRUNCSTS.x, x = UIE, URE, MTLPE, UCE, CAE, CTE, PTLPE | |
| PMCSR.PMEEN | |
| RID.RID | |
| **Primary-Reset-Only (PRST), Including Read-Write-Once (RW-O)** | |
| AERCAPHDR.NCO | |
| CAPPTR.CPTR | |
| CCR.x, x = BASE, SUB, RLPI | |
| DEVCAP.FLR | |
| DEVCAP2.CTRS | |
| INTP.INTP | |
| PLKCTL.CL | |
| PMCSR.NSR | |
| SID.SID | |
| SVID.SVID | |

Exceptions to function-level resets are listed in Table 29-25.

**Table 29-25. SMT Function Level Reset Exceptions**

| Register Fields | Comments |
|---|---|
| DEVCTL.x, x = IFLR and MPS | In addition to any RWS, ROS, and RW1CS fields. |
| IOSFDEVCLKGCTL.x, x = ICGE and ICT | |
| SBDEVCLKGCTL.x, x = ICGE and ICT | |

## 29.6.2 Target Reset (GCTRL.TRST)

The scope of this reset is limited to the target FSM. Upon assertion, the hardware gracefully terminates any pending transaction at the next byte boundary and builds a descriptor which the firmware rejects.

## 29.7 MCTP Over SMBus Packet Header Format

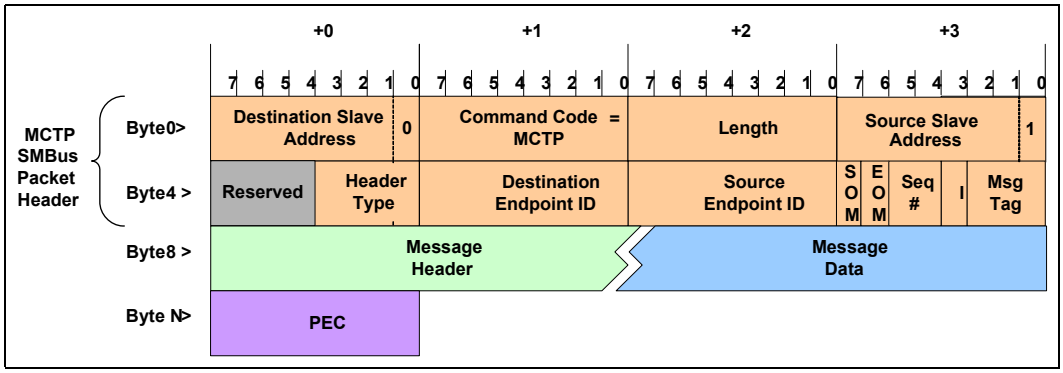
This section is informational. The hardware does not inspect or decode any portion of an Management Component Transport Protocol (MCTP) packet.

MCTP messages with SMBus packet headers can encapsulate data from various protocols like ASF, Network, etc. The command code in the packet (byte 1) identifies this packet as MCTP and the firmware takes appropriate action. Refer to the *Management Component Transport Protocol (MCTP) Base Specification, Version 1.1.0* for further details on the protocol.

The following information is contained in the *Management Component Transport Protocol (MCTP) Base Specification, Version 1.1.0* but is reproduced here for completeness.

https://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.1.0.pdf

**Figure 29-15.MCTP Over SMBus Packet Format**



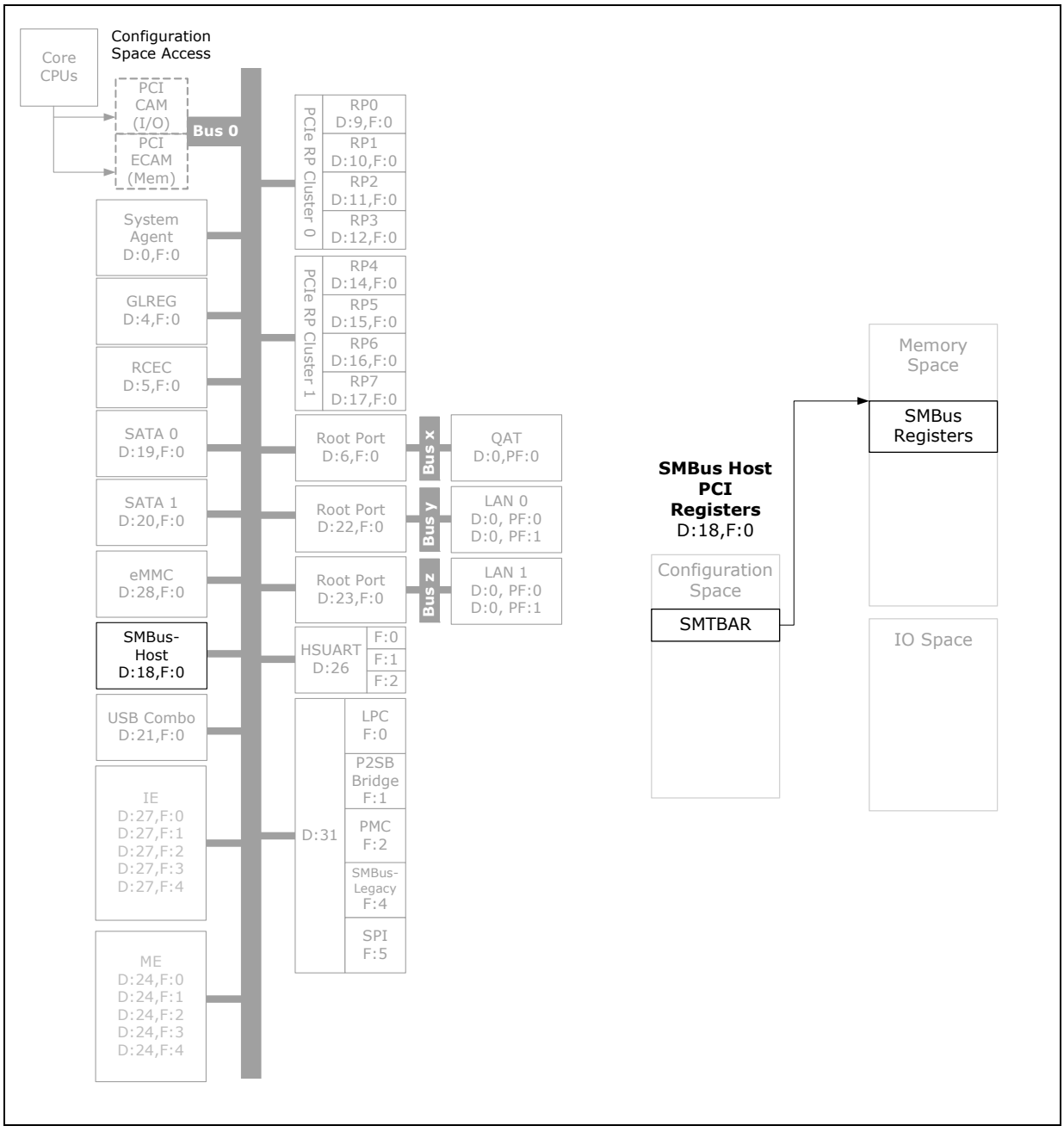
## Table 29-26. MCTP Over SMBus Packet Format

| Byte # | Field | Description |
|---|---|---|
| 0 | Destination Slave Address | For the local SMBus, slave address for the target device.<br>Bit 0 is always 0 since all MCTP messages are writes. |
| 1 | Command Code | Command code indicating the MCTP packet. This value is fixed at 0Fh. |
| 2 | Length | Length of the block write. Analogous to the byte count field of the SMBus Block write except length is not limited to 32 bytes. This count indicates the number of bytes to follow in the current message without counting the PEC byte. |
| 3 | Slave Address | Bits [7:1]: For the local SMBus, the slave address of the message initiator. This initiator is a repeater and cannot be assumed to be the original initiator.<br>Bit 0: Reserved, always 1b. |
| 4 | Header Type | Bits [7:4]: Reserved<br>Bits [3:0]: MCTP Version: 1h for all MCTP device compliant to the *Management Component Transport Protocol (MCTP) Base Specification,* Version 1.0.0. |
| 5 | Destination Endpoint ID | Unique ID on the system for the destination of the message. This ID reflects the final destination of the message.<br>Reserved Values:<br>0 = Local bus only. Terminate at the receiver.<br>1 = Route to the root management controller.<br>2-7 = Reserved for future definition. |
| 6 | Source Endpoint ID | Unique ID on the system for the originator of the message. This ID reflects the true initiator of the message. |
| 7 | Message Control | Bit 7: **Start of Message** (SOM): Set to 1 if the packet is the start of message.<br>Bit 6: **End of Message** (EOM): Set to 1 if the packet is the end of message.<br>Combinations of SOM and EOM are used to identify the start, middle, and end of a message that is split across one or more packets.<br>Bits [5:4]: **Sequence Number**: For messages that span multiple packets, this field helps identify lost intermediate packets. Increments on each successive packet. The value of the sequence number for Packet N, is the Packet N-1 Sequence Number + 1 mod 4.<br>Bit 3: **Initiator** (I): Set to 1 if the value of the message tag field was assigned by the source (for requests) or set to 0 if the message tag was assigned by the destination (for responses).<br>Bits [2:0]: **Message Tag** (MT): This field, along with the source endpoint ID, identifies a unique message ID. The message ID must be unique for all outstanding request transactions that require a response and have not yet received the response. For response messages, the message tag sent with the request is returned in the response. |
| 8 | Message Header | Begin MCTP message header. |
| M | Message Data | MCTP defined |
| N | PEC | Single byte packet error check accompanies all MCTP messages |

## 29.8 Register Map

Figure 29-16 shows the associated registers from a system software viewpoint.

**Figure 29-16.Register Map**




§ §

# 30 SMBus Controller - Platform Environment Control Interface

The Platform Environment Control Interface (PECI) was developed to replace $I^2C$ as the methodology of reading CPU temperatures. The PECI specification has evolved overtime to provide a broader management interface to manage the platform. In non-SoC environments, the PECI slave interface is implemented in the CPU complex, while the master interface is in the Platform Control Hub (PCH) and/or Baseboard Management Controller (BMC). The PECI implementation only allows a single master on the bus.

On the SoC platform, the BMC/IE acts as a PECI over SMBus master and the SoC SMBus-for-PECI controller described in this chapter acts as a slave PECI proxy controller. The PECI commands are encapsulated within SMBus packets sent to the SoC. Additional, new PECI commands have been defined for the SoC, specific to its architecture.

**Figure 30-1. What is Covered in This Chapter**

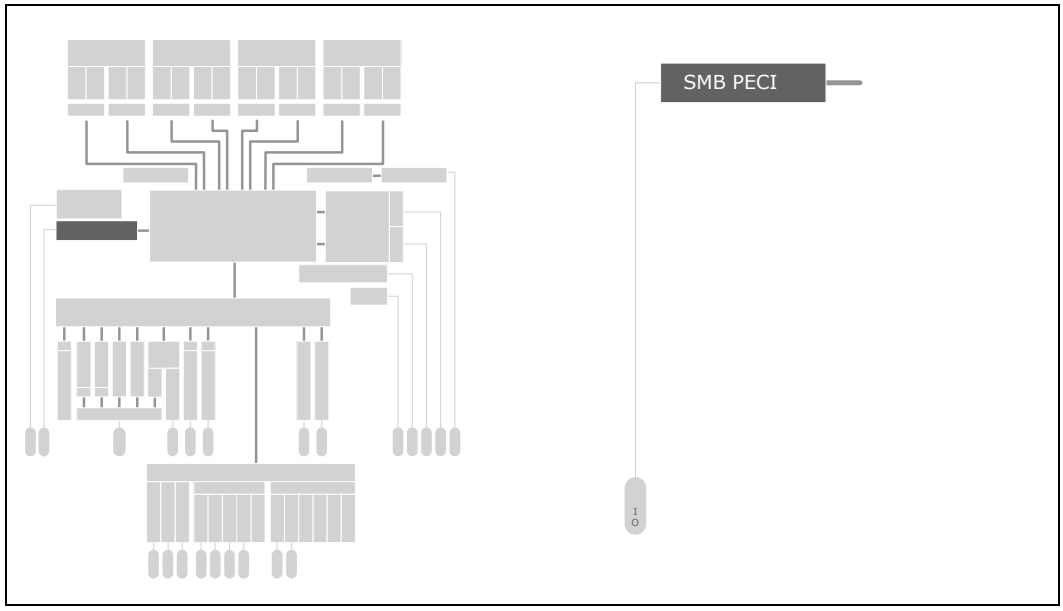



**Table 30-1. References**

| Document Title | Document ID / Location |
|---|---|
| *System Management Bus (SMBus) Specification, Version 3.0* | http://smbus.org/specs/SMBus_3_0_20141220.pdf |

## 30.1 Signal Descriptions

The signal descriptions are shown in Table 30-2, "Signal Names." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a customer GPIO signals, see Section 39.1, "Directory of Signal Names and Pin Names." The Direction/Type column of Table 30-2, "Signal Names" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.

**Table 30-2. Signal Names**

| Signal Name | Direction Type | Shared | Description |
|---|---|---|---|
| SMB_PECI_CLK | I, O-OD | Yes | SMBus Clock (SMBCLK) |
| SMB_PECI_DATA | I, O-OD | Yes | SMBus Data (SMBDAT) |

The optional SMBus 2.0 signals, SMBALERT# and SMBSUS#, are not supported on this controller interface.

## 30.2 PECI over SMBus Features

The SMBus supports the external BMC or IE, operates in slave mode only and comprehends the PECI command structure.

The PECI processes one command at time and support for multiple outstanding commands is not supported.

The PECI commands supported by the SoC are in Table 30-7, "Supported PECI Commands."

## 30.3 SMBus Supported Transactions

In the SoC, the communication between the SMBus controller and the internal PECI bridge utilizes the SMBus block write and block read transactions as defined in the SMBus 2.0 specifications.

The SMBus PECI device is assigned two SMBus Slave Addresses. The two Slave Addresses are assigned default values but can be altered by the customer Soft Straps. See the PECI category in Section 4.2, "SoC Soft Straps."

- 4Bh = Default SMBus Slave Address for all SMBus Read transactions
- 4Ch = Default SMBus Slave Address for all SMBus Write transactions

The SMBus Slave Addresses are located in sideband registers.

- for SMBus Read transactions: GPTRADR[7:0] field of the GPBRCTRL register located at sideband Port CEh, offset 240h
- for SMBus Write transactions: ADDR0[7:1] field of the TACTRL register located at sideband Port CEh, offset 228h

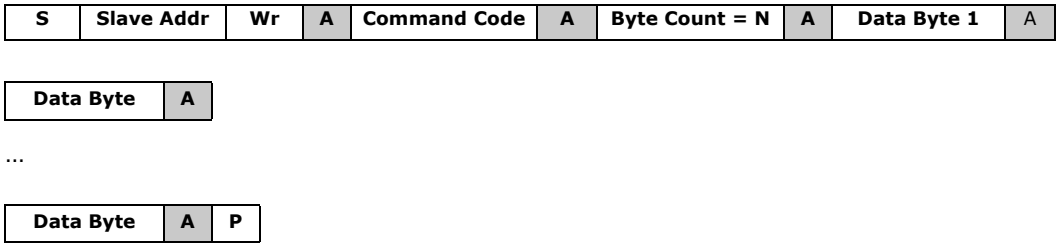
See Figure 30-2, "SMBus Protocol" to interpret the SMBus protocol drawing.

**Figure 30-2. SMBus Protocol**

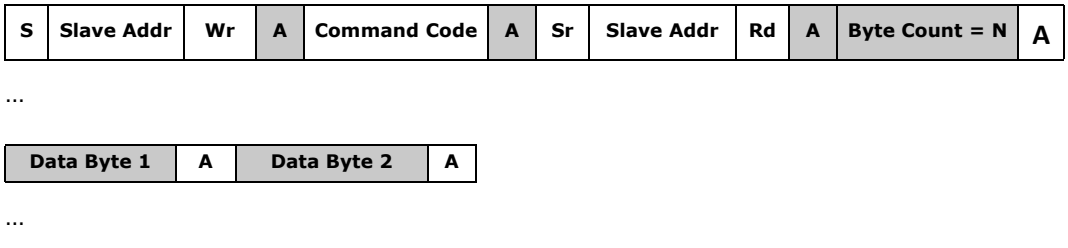
| | |
|---|---|
| S | Start Condition |
| Sr | Repeated Start Condition |
| Rd | Read (bit value of 1) |
| Wr | Write (bit value 0) |
| X | Shown under a field indicates that field is required To have the value of 'x' |
| A | Acknowledge (this bit position is 0 for an ACK or 1 for a NACK) |
| P | Stop Condition |
| PEC | Packet Error Code |
| … | Continuation of Protocol |
| | Master to Slave |
| | Slave to Master |

## 30.4 SMBus Block Read/Write Transaction Formats

### Figure 30-3. SMBus Block Write Command




### Figure 30-4. SMBus Block Read Command




## 30.5 SMBus Commands

### Table 30-3. SMBus Write Commands

| Operating Mode | Transaction | Slave Address | Command Code | Byte Count (N) | Byte 1 | Byte [N:1] |
|---|---|---|---|---|---|---|
| PECI Mode | PECI Proxy Command | 0x4C | 0x62 | 0x4...N[1] | | |
| PECI Mode | Reset PECI | 0x4C | 0x64 | 0 | | |

1. See Section 30.6.2.1, "PECI Proxy Command Format."

### Table 30-4. SMBus Read Command

| Operating Mode | Transaction | Slave Address (CMD Phase) | Read Command Code | Slave Address (Data Phase) | Byte Count (N) | Bytes [N:1] |
|---|---|---|---|---|---|---|
| PECI Mode | PECI Proxy Read Command | 0x4B | 0x40 | 0x4B | 0x01...N[1] | |

1. See Section 30.6.2.2, "PECI Proxy Read Command."

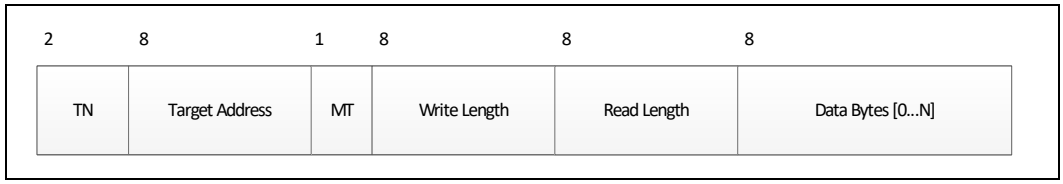
## 30.6 PECI Over SMBus

PECI uses a simple structure of message header and a write-read protocol. All PECI devices have a command field. The following section describes PECI protocol.

### 30.6.1 PECI Message Header in SMBus

The header conveys to the target device how many bytes the master intends to send and how many it expects to receive back. The first byte of the write data is interpreted as a command to the device and must be present in all messages. The Ping() command is the only exception to this rule. Additional bytes are written to convey sub commands or to send data to the device. A zero value in the read length field means no data is read from the target device. See Figure 30-5, "PECI Message Header in the SMBus Packet" for the PECI message header. In the figure, N bytes are to be written to the target and M bytes are to be read back from the target.

**Figure 30-5. PECI Message Header in the SMBus Packet**




In the PECI message header, the Address Timing Negotiation (NT) and Message Timing Negotiation (MT) bits are not present. The BMC is responsible to not include these timing bits in the header.

#### 30.6.1.1 Target Address Field

The PECI device address is defined as 0x30 + a socket identifier of 0…7. In the SoC, only a single socket model is supported, thus the PECI target address is always 0x30. Other PECI device addresses are rejected as errors.

#### 30.6.1.2 Write Length Field

The Write Length byte conveys the number of bytes the originator sends the target device. Since all PECI message headers are identical in size, with the exception of Ping(), the Write Length byte only describes the number of command bytes sent. The target address, write length and read length fields are not included in the write length count. The following rules apply for the write length field:

- The maximum value is 0xFF.
- The command byte is included in the length.
- A write length of 0x0 is only for the Ping() command.

#### 30.6.1.3 Read Length Field

The Read Length byte conveys the number of bytes the target device must supply the originator before returning the Frame Check Sequence (FCS) byte calculated over that data. The read length is in the range of 0x00 and 0xFF.

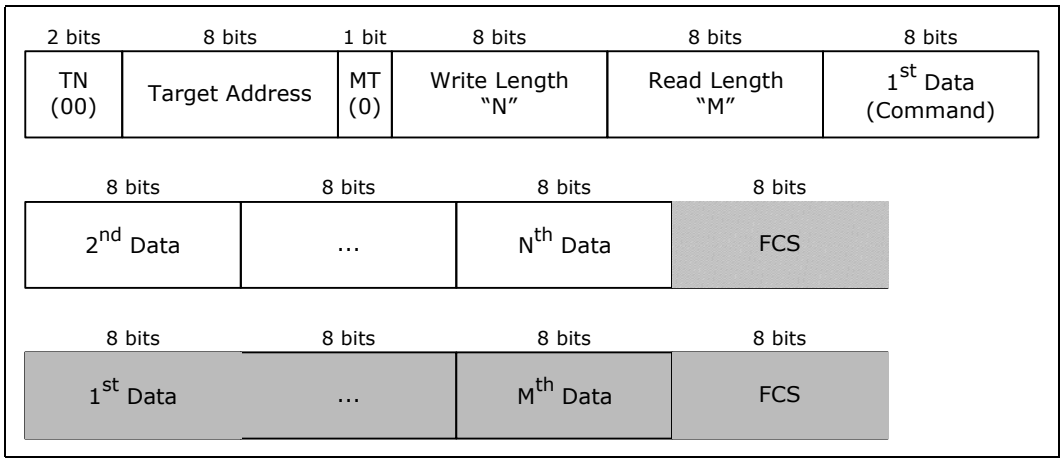
#### 30.6.1.4 Command Byte

All SoC PECI commands except Ping() require a command byte. The command code is positioned as the first data byte, that is, data_byte[0].

## 30.6.2 PECI Write-Read Protocol

The write-read protocol shown in Figure 30-6, "PECI Write-Read Protocol" is the only protocol defined for messaging between devices on the PECI. The write-read protocol allows an atomic operation that first writes and then reads data between an originator and a target. Format of a write-read message is given in Figure 30-6, "PECI Write-Read Protocol". The read and write Frame Check Sequence (FCS) bytes are not calculated but may be required to have a byte allocated as part of the specific command where described in Section 30.6.2.1, "PECI Proxy Command Format."

**Figure 30-6. PECI Write-Read Protocol**




Legend:

Master to Slave

Slave to Master

Similar to Figure 30-5, "PECI Message Header in the SMBus Packet," the PECI message header, the Address Timing Negotiation (NT) and Message Timing Negotiation (MT) bits are not present. The BMC is responsible to not include these timing bits in the header.

### 30.6.2.1 PECI Proxy Command Format

Unlike some Intel Platform Control Hub (PCH) products, the SoC PECI-over-SMBus controller only operates in PECI mode. The SMBus Interface PECI Proxy Command protocol is an encapsulation of PECI specification commands onto the SMBus by the SMBus master. Two new SoC-specific PECI commands are also supported.

The PECI proxy command bytes, encapsulated in bytes [N:2] of the protocol, conform to Figure 30-6, "PECI Write-Read Protocol". The SoC PECI-over-SMBus controller provides a transparent pass-through of the PECI command to the integrated PECI client and does not verify the validity of the command. It is the burden of the BMC or platform board External Circuitry (EC) to provide valid data in the PECI proxy command fields of the incoming packet.

Because the SMBus is used and not the serial PECI electrical interface, it is unnecessary to calculate the Frame Check Sequence (FCS). The SoC design does keep the PECI commands consist with non-SoC Intel products and carry additional bytes that are set to 0x0. These are called out where appropriate. The PECI proxy command format is shown in Table 30-5, "PECI Proxy Command Protocol Format."

**Table 30-5. PECI Proxy Command Protocol Format**

| PECI Proxy Command | | | | | | | |
|---|---|---|---|---|---|---|---|
| SMBus Default Data | | | SMBus to PECI Handshake Control Data | PECI Command Data as Described for Each Supported PECI Command in the PECI Specification | | | |
| Slave Address | PECI Mode Command Code | Byte Count | Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte [N:5] |
| 0x4C (default) | 0x62 | 0x04...N | Control Byte (Reserved) | PECI Target Address[1] 0x30 = CPU | PECI Write Length[2] | PECI Read Length[3] | Remaining Part of PECI Command[4] |

1. Required PECI header byte.
2. Required PECI header byte. Must be set to the proper value as defined by the particular PECI command. If an Assured Write FCS (AW FCS) is needed, the PECI write length value must also include the AW FCS byte.
3. Required PECI Header byte. Must be set to the proper value as defined by the particular PECI command.
4. This field does not exist for the PECI Ping() command. This field is used for write data bytes including a placeholder for an AW FCS byte. Note that the Retry bit shall be set to zero and the command code byte must be one of the codes supported by the SoC.

### 30.6.2.2 PECI Proxy Read Command

The SMBus Read command uses one SMBus command code (0x40). The SMBus Read command returns PECI command status or PECI response payload data from the executed PECI Proxy command.

Table 30-6, "PECI Proxy Read" shows the structure of PECI response data from a PECI mode read.

**Table 30-6. PECI Proxy Read (Sheet 1 of 2)**

| SMBus Field | Value | Data Source | Description |
|---|---|---|---|
| Slave Address | 0x4B | BMC | Slave address for the command phase |
| Command Code – SMBus Read | 0x40 | BMC | Command code for SMBus read |
| Slave Address | 0x4B | BMC | Slave address for data phase |

## Table 30-6. PECI Proxy Read (Sheet 2 of 2)

| SMBus Field | Value | Data Source | Description |
|---|---|---|---|
| Byte Count (N) | 0x01...N | SoC | N = 0x01 for busy state:<br>• Byte 1 = 0x01 (CMD_BUSY bit set)<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02 (CMD_ERR bit set)<br>• Byte 2 = Error code<br>N > 2 indicates the request PECI command executed successfully:<br>• Byte 1 = 0x0<br>• Byte 2 = 0x0<br>• Byte [N:3] = PECI response data |
| Byte 1 | Control Data | | **Status Byte (CMD_STAT):**<br>Bits [7:2] = Reserved<br>Bit 1 = CMD_ERR<br>• This bit is set by the SoC if the PECI proxy interface detects a transaction error. The error code is defined in Byte 2.<br>Bit 0 = CMD_BUSY<br>• This bit is set by the SoC while a PECI command is executing. It is reset by the SoC when the PECI command operation is completed. |
| Byte 2 | Control Data | | **Error Code (ERR_CODE):**<br>0x00 = PECI_ST_OK<br>• PECI command completed successfully<br>0x01 = PECI_ST_BAD_FORMAT<br>• SMBus PECI message incorrectly formatted<br>0x09 = PECI_ST_DISABLED<br>• PECI driver disabled<br>0x3E = PECI_INVALID_SOCKET<br>• Invalid PECI socket number<br>0xE1 = PECI_ST_TRANS_TIMEOUT_ERROR<br>• PECI transaction time-out (e.g., PECI controller not responding to commands or hangs)<br>0xFE = PECI_ST_FAILURE<br>• General PECI failure<br>0xFF = PECI_ST_UNKNOWN<br>• Unknown status<br><br>**Unsupported Error Codes:**<br><br>0x02 = PECI_ST_FCS_REQ_ERROR<br>• Error in PECI request FCS<br>0x04 = PECI_ST_FCS_RSP_ERROR<br>• Error in PECI response FCS<br>0x08 = PECI_ST_LINK_ERROR<br>• PECI link error |
| Byte [N:3] | PECI Response Data | | **PECI Response Data:**<br>PECI response data as described for each PECI command in the PECI specification. It contains all data returned by the PECI client that resides between the Write FCS byte and Read FCS byte.<br>These bytes are valid only when the PECI command completes successfully and the CMD_ERR bit in the Status byte (Byte 1; Bit [1]) is cleared. |

### 30.6.3 PECI Proxy Command Handling Procedure

A BMC requests read and write PECI proxy commands over the SMBus that are formatted to conform to the PECI command protocol. The PECI command handler performs the following on receipt of the SMBus packet:

- Validates that the PECI command is valid.
- Either processes the command locally or forwards the command to other SoC internal units for processing.
- Repackages the command into the appropriate format upon command completion.

The BMC retrieves the PECI response data by performing the SMBus block read transactions. The number of data bytes returned from the Block Read command on the SMBus indicates the completion status of the PECI command.

1. If N=1, then only the Status byte is sent with CMB_BUSY bit set (CMD_BUSY=1b).
   Indicates that the PECI command transaction is still in progress.
2. If N=2, then both Status and Error Code bytes are sent with CMD_BUSY bit reset (CMD_BUSY = 0b) and CMD_ERR bit set (CMD_ERR = 1b).
   Indicates that the PECI command resulted in an error. Byte 2 contains the error code.
3. If N>2, then both CMD_BUSY and CMD_ERR will be reset (CMD_BUSY = 0b, CMD_ERR = 0b).
   Indicates that the PECI command completed successfully and the read-back data is valid.

*Note:* The BMC should only trigger a new PECI command when the previous command is completed. The SoC does not support posted or multiple commands.

*Note:* If the Reset PECI SMBus command is received, any pending state will be reset. The initiator would be expected to reissue the last command.

### 30.6.4 PECI Proxy Command Trigger

This process provides the functions performed by the SMBus master (the BMC) and the SoC to trigger and complete a PECI Proxy command.

1. The BMC performs an SMBus block write transaction, formatted with the data associated with the requested PECI proxy command as shown in Table 30-6, "PECI Proxy Read."
2. The SoC sets the CMD_BUSY bit in the Status byte and handles the requested PECI command.
3. After completing the PECI operation, the SoC performs the following functions:
   — Resets the CMD_BUSY bit in the Status byte.
   — Sets the CMD_ERR bit if a PECI transaction error occurred; otherwise, it resets CMD_ERR bit.
   — Sets the ERR_CODE byte with the end of transaction condition.
   — If the PECI transaction completed successfully, stores the PECI response data and transfers it to the BMC on the next Read command.
4. The BMC polls the Status byte until the CMD_BUSY bit is cleared to indicate the completion of the command. If the PECI transaction completed successfully (CMD_ERR = 0), the PECI response data is valid.

#### 30.6.4.1 Unsupported PECI Command

In the case of an unsupported PECI command, the SoC responds with the appropriate error code.

#### 30.6.4.2 Illegally Formatted Command

In the case of an unsupported PECI command, the SoC responds with the appropriate error code.

## 30.7 PECI Proxy Commands

Table 30-7, "Supported PECI Commands" is a summary of the PECI Proxy commands supported by the SoC. Each is described in this section following the table.

**Table 30-7. Supported PECI Commands**

| Command | In PECI Specification, Rev. 3.1 | Supported by SoC | Command Code | Definition |
|---|---|---|---|---|
| Ping() | Yes | Yes | none | This command detects that PECI functionality exists. |
| GetDIB() | Yes | Yes | 0xF7 | This command returns the PECI device-specific information. |
| GetTemp() | Yes | Yes | 0x01 | This command returns the processor die information. |
| RdPkgConfig() | Yes | Yes | 0xA1 | This command reads the SoC package configuration space. |
| WrPkgConfig() | Yes | Yes | 0xA5 | This command writes the SoC package configuration space. |
| RdIAMSR() | Yes | Yes | 0xB1 | |
| WrIAMSR() | Yes | No | - | |
| RdPCIConfig() | Yes | No | - | |
| WrPCIConfig() | Yes | No | - | |
| RdPCIConfigLocal() | Yes | Yes | 0xE1 | This command reads the SoC PCI space. |
| WrPCIConfigLocal() | Yes | Yes | 0xE5 | This command writes the SoC PCI space. |
| RdEndPointConfig() | No | Yes | 0xC1 | This command reads the register space over the sideband. |
| WrEndPointConfig() | No | Yes | 0xC5 | This command writes the register space over the sideband. |

### 30.7.1 Ping()

Ping() is a required message for all PECI devices. This message is used to enumerate devices or determine if a device has been removed, been powered-off, etc. A Ping() sent to a device address always returns a non-zero write FCS if the device at the targeted address is able to respond.

Table 30-8, "Ping - PECI Proxy Block Write" shows the ping PECI proxy block write format and Table 30-9, "Ping - PECI Proxy Block Read" shows the ping PECI proxy block read format.

**Table 30-8. Ping - PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x04 | | |
| Byte 1 | SMBUS-PECI Handshake Control | | 0x00 | | |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | 0x00 | | Ping() command has no command code.<br>• Write Length = 0x00<br>• Read Length = 0x00 |
| Byte 4 | PECI Read Length | | 0x00 | | |

### Table 30-9. Ping - PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBUS Read Command Code | SMBUS Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for PECI client status<br>• Byte 1 = 0x00<br>If Byte 2 = 0x00<br>• PECI client active<br>If Byte 2 = 0x02<br>• PECI client not active |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] of Table 30-6, "PECI Proxy Read.". |
| Byte 2 | Error Byte | | ERR_CODE | | If Byte 2 = 0x00<br>• PECI client active<br>If Byte 2 = 0x02<br>• PECI client not active.<br>See Byte[2] of Table 30-6, "PECI Proxy Read" for error code definitions. |

## 30.7.2 GetDIB()

The SoC PECI client implementation of GetDIB() includes an 8-byte response and provides information regarding client revision number and the number of supported domains.

Table 30-10, "GetDIB() PECI Proxy Block Write" shows the GetDIB PECI proxy block write format and Table 30-11, "GetDIB() PECI Proxy Block Read" shows the GetDIB PECI proxy block read format.

**Table 30-10. GetDIB() PECI Proxy Block Write**

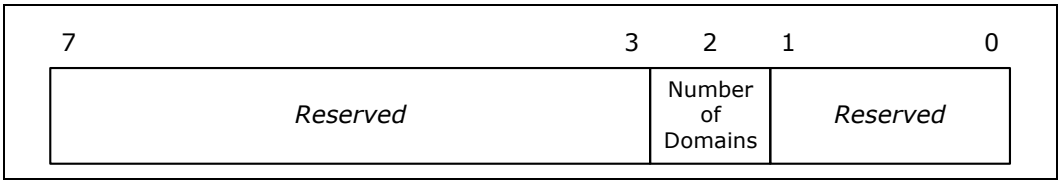
| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x05 | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | Socket ID. Always 0x30 for the SoC. |
| Byte 3 | PECI Write Length | | 0x01 | | |
| Byte 4 | PECI Read Length | | 0x08 | | |
| Byte 5 | GetDIB() Command Code | | 0xF7 | | |

### Table 30-11. GetDIB() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x0A} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for Transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined in Table 30-6, "PECI Proxy Read."<br>N = 0xA indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2= 0x00<br>• Byte[10:3] = PECI response data |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] of Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte[2] of Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Response Byte 1 | PECI Response Data | PECI Device Info | | See Section 30.7.2.1, "PECI Device Info Field."<br>Returns the number of domains in addition to Domain#0 as shown below. |
| Byte 4 | PECI Response Byte 2 | | PECI Device Revision | | See Section 30.7.2.2, "PECI Revision Number." |
| Byte[10:5] | PECI Response Byte [8:3] | | Reserved | | |

### 30.7.2.1 PECI Device Info Field

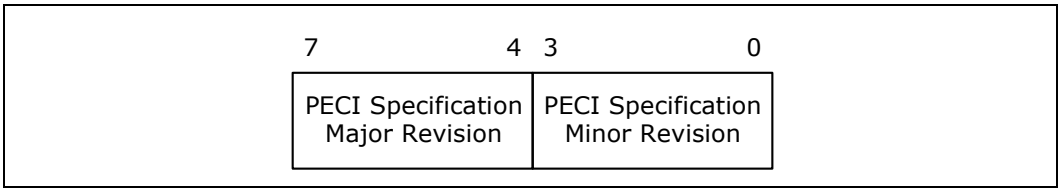
**Figure 30-7. PECI Device Info Field Definition**




### 30.7.2.2 PECI Revision Number

The revision number returned is 0011_0001b.

**Table 30-12. PECI Revision Number Definition**



### 30.7.3 GetTemp()

The GetTemp() command is used to retrieve the temperature from a target PECI address. The temperature is used by the BMC to regulate the temperature on the die. The data is returned as a negative value representing the number of degrees Celsius below the maximum processor Junction Temperature ($T_{J\text{-}MAX}$).

*Note:* The maximum PECI temperature value of zero corresponds to the processor $T_{J\text{-}MAX}$. This also represents the default temperature at which the processor Thermal Control Circuit (TCC) activates. The actual value that the thermal management system uses as a control set point ($T_{CONTROL}$) is also defined as a negative number below $T_{J\text{-}MAX}$.

See Section 30.11, "DTS Temperature Data" for the data format and error codes. Table 30-13 shows the GetTemp PECI proxy block write format and Table 30-14, "GetTemp() PECI Proxy Block Read" shows the GetTemp PECI proxy block read format.

**Table 30-13. GetTemp() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x05 | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | 0x01 | | |
| Byte 4 | PECI Read Length | | 0x02 | | |
| Byte 5 | GetTemp() Command Code | | 0x01 | | |

## Table 30-14. GetTemp() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x04} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined in Table 30-6, "PECI Proxy Read."<br>N = 0x4 indicates successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2= 0x00<br>• Byte [4:3] = PECI response data |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | CMD_ERR | | See Byte[2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Response Byte 1 | PECI Response Data | PECI Device Temp [7:0] | | See Section 30.11.1, "PECI Device Temp Data." |
| Byte 4 | PECI Response Byte 2 | | PECI Device Temp [15:8] | | |

## 30.7.4 RdPkgConfig()

The RdPkgConfig() command provides read access to the Package Configuration Space (PCS) within the processor including various power and thermal management functions. Typical PCS read services supported by the processor may include access to temperature data, energy status, run time information, DIMM temperatures and so on.

This command does provide multi-domain support. Table 30-15, "RdPkgConfig() PECI Proxy Block Write" shows the RdPkgConfig PECI proxy block write format and Table 30-16, "RdPkgConfig() PECI Proxy Block Read" shows the RdPkgConfig PECI proxy block read format.

**Table 30-15. RdPkgConfig() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x09 | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | 0x05 | | |
| Byte 4 | PECI Read Length | | {0x02, 0x3, 0x5} | | |
| Byte 5 | RdPkgConfig() Command Code | | 0xA1 | | |
| Byte 6 | Host ID and Retry | | 0x00 | | |
| Byte 7 | Index | | | | See Section 30.7.11, "MCA Access via RdEndPointConfig()" and Section 30.9, "CPU Thermal and Power Optimization Capabilities" for the supported capabilities and their index and parameter information. |
| Byte 8 | Parameter LSB | | | | |
| Byte 9 | Parameter MSB | | | | |

## Table 30-16. RdPkgConfig() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x04, 0x05, 0x07} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined in Table 30-6, "PECI Proxy Read."<br>N = {0x4,0x5,0x7} indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte [7:3] = PECI response data<br>N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte [7:4] on the SMBus.<br>N Value:<br>• 0x04 = 1 data byte<br>• 0x05 = 2 data bytes<br>• 0x07 = 4 data bytes |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte [1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte [2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion Code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |
| Byte 4 | Data 1 (LSB) | | Data = 1, 2, or 4 bytes | | See Section 30.7.11, "MCA Access via RdEndPointConfig()" and Section 30.9, "CPU Thermal and Power Optimization Capabilities" for the supported capabilities and their returned data structures. |
| Byte 5 | Data 2 | | | | |
| Byte 6 | Data 3 | | | | |
| Byte 7 | Data 4 (MSB) | | | | |

## 30.7.5 WrPkgConfig()

The WrPkgConfig() command provides write access to the Package Configuration Space (PCS) within the processor including various power and thermal management functions. Typical PCS write services supported by the processor may include power limiting, thermal averaging constant programming, and other write services.

Table 30-17, "WrPkgConfig() PECI Proxy Block Write" shows the WrPkgConfig PECI proxy block write format and Table 30-18, "WrPkgConfig() PECI Proxy Block Read" shows the WrPkgConfig PECI proxy block read format.

**Table 30-17. WrPkgConfig() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | {0x0B, 0x0C, 0x0E} | | N = {0x0b, 0x0c, 0x0e} value is determined by the number of PECI Write Data bytes transmitted in Byte [13:10] on the SMBus.<br>N Value:<br>• 0x0b = 1 data byte<br>• 0x0c = 2 data bytes<br>• 0x0e = 4 data bytes |
| Byte 1 | SMBus-PECI Handshake Control | | 0x01 | | Value = 0x01<br>• Assured Write (AW) Frame-Check Sequence (FCS) is not required. However, to maintain compatibility included is the AW space for the byte per the PECI specification. |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | {0x07, 0x08, 0x0A} | | |
| Byte 4 | PECI Read Length | | 0x01 | | |
| Byte 5 | WrPkgConfig() Command Code | | 0xA5 | | |
| Byte 6 | Host ID and Retry | | 0x00 | | |
| Byte 7 | Index | | | | See Section 30.7.11, "MCA Access via RdEndPointConfig()" and Section 30.9, "CPU Thermal and Power Optimization Capabilities" for the supported capabilities and their index, parameter, and data structure information. |
| Byte 8 | Parameter (LSB) | | | | |
| Byte 9 | Parameter (MSB) | | | | |
| Byte 10 | PECI Data 1 (LSB) | | | | |
| Byte 11 | PECI Data 2 | | | | |
| Byte 12 | PECI Data 3 | | | | |
| Byte 13 | PECI Data 4 (MSB) | | | | |
| Byte 14 | AW FCS Byte | | 0x00 | | Dummy to maintain compatibility with the PECI specification. |

## Table 30-18. WrPkgConfig() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x03} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined in Table 30-6, "PECI Proxy Read".<br>N = {0x03} indicates successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte3 = PECI Completion code |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte [1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte [2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |

## 30.7.6 RdIAMSR()

The RdIAMSR() PECI command provides read access to Model Specific Registers (MSRs) defined in the processor's Intel® Architecture (IA). MSR definitions can be found in the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

The processor PECI client allows PECI RdIAMSR() access to the registers pertaining to the SoC error banks (machine check banks 0 through 19). The exact number of accessible banks for the SoC may be obtained by reading the IA32_MCG_CAP[7:0]MSR (0x0179). Attempts to read MSRs that are not accessible over PECI or not implemented result in a completion code of 0x90.

PECI access to these registers is expected only when in-band access mechanisms are not available. Valid processor ID values may be obtained by using the enumeration methods described in Section 30.7.6.1, "Processor ID Enumeration." The PECI host must determine the total number of machine check banks and the validity of the MCi_ADDR and MCi_MISC register contents prior to issuing a read to the machine check bank similar to standard machine check architecture enumeration and accesses. The processor machine check banks 4 through 21 reside outside the processor and hence will return the same value independent of the processor ID used to access these banks. PECI accesses to the machine check banks may not be possible in the event of a core hang. A warm reset of the processor may be required to read any sticky machine check banks.

### 30.7.6.1 Processor ID Enumeration

The 'Processor ID' field that is used to address the IA MSR space refers to a specific logical processor within the CPU. The 'Processor ID' always refers to the same physical location in the processor silicon regardless of configuration. For example, if certain logical processors are disabled by BIOS, the Processor ID mapping will not change. The total number of Processor IDs on a CPU is product-specific. Processor ID does not always refer to the same physical location across processor types and SKUs.

'Processor ID' enumeration involves discovering the logical processors enabled within the CPU package. This can be accomplished by reading the 'Max Thread ID' value through the RdPkgConfig() command (Index 0, Parameter 3) and subsequently querying each of the supported processor threads. Unavailable processor threads will return a completion code of 0x90.

Table 30-19, "RdIAMSR() PECI Proxy Block Write" shows the RdIAMSR PECI proxy block write format.

Table 30-20, "RdIAMSR() PECI Proxy Block Read" shows the RdIAMSR PECI proxy block read format.

## Table 30-19. RdIAMSR() PECI Proxy Block Write

| SMBUS | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBUS Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x09 | | |
| Byte 1 | SMBUSPECI Handshake Control | | 0x00 | | Value = 0x00 No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | 0x05 | | |
| Byte 4 | PECI Read Length | | {0x02, 0x3, 0x5, 0x9} | | |
| Byte 5 | RdIAMSR () Command Code | | 0xB1 | | |
| Byte 6 | Host ID & Retry | | 0x00 | | |
| Byte 7 | Processor ID | | | | See supported Structure sections for details. |
| Byte 8 | MSR Address (LSB) | | | | |
| Byte 9 | MSR Address (MSB) | | | | |

***Note:*** The "Processor ID" field used to address the IA MSR space refers to a specific logical processor in the CPU. The Processor ID always refers to the same physical location in the processor silicon regardless of configuration. For example, if certain logical processors are disabled by BIOS, the Processor ID mapping will not change. The total number of Processor IDs on an SoC is product-specific.

## Table 30-20. RdIAMSR() PECI Proxy Block Read

| SMBUS | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBUS Command | 0x4B | BMC | |
| SMBUS Read Command Code | SMBUS Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x04, 0x05, 0x07, 0x0b} | PECI "Command Controller" | N = 0x01 for Busy state<br>• Byte 1 = 0x01<br>N = 0x02 for Transaction Errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined Table 30-6, "PECI Proxy Read.".<br>N = {0x4,0x5,0x7, 0x0b} indicates successful PECI Transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte[11:3] = PECI Response Data<br>N = {0x04, 0x05, 0x07,0x0b} value is determined by the number of PECI Response Data bytes returned in Byte[11:4] on the SMBus.<br>N Value:<br>• 0x04 = 1 data byte<br>• 0x05 = 2 data bytes<br>• 0x07 = 4 data bytes<br>• 0x0b = 8 data bytes |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] of Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte[2] of Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion Code decode:<br>0x40 = Command successful<br>0x80 = Response time-out<br>0x90 = Illegal command |
| Byte 4 | Data 1 (LSB) | | Data = {1, 2, 4, or 8 bytes) | | See later section. |
| Byte 5 | Data 2 | | | | |
| Byte 6 | Data 3 | | | | |
| Byte 7 | Data 4 | | | | |
| Byte 8 | Data 5 | | | | |
| Byte 9 | Data 6 | | | | |
| Byte 10 | Data 7 | | | | |
| Byte 11 | Data 8 (MSB) | | | | |

### 30.7.7 RdPCIConfigLocal()

The RdPCIConfigLocal() command provides sideband read access to the entire PCI configuration space of the SoC. This includes all SoC registers within the PCI configuration space as described in register document.

PECI originators may conduct a device/function enumeration sweep of this space by issuing reads in the same manner that the BIOS would. A response of all 1's may indicate that the device/function/register is unimplemented even with a 'passing' completion code. Alternatively, reads to unimplemented or hidden registers may return a completion code of 0x90 indicating an invalid request. It is also possible that reads to function 0 of non-existent IIO devices issued prior to BIOS POST may return all '0's with a passing completion code. PECI originators can access this space even prior to BIOS enumeration of the system buses. There is no read restriction on accesses to locked registers.

PCI configuration addresses are constructed as shown in Figure 30-8, "PCI Configuration Address Format." Under normal inband procedures, the Bus number would be used to direct a read or write to the proper device. Any request made with a bad Bus number is ignored and the client will respond with all '0's and a 'passing' completion code.

**Figure 30-8. PCI Configuration Address Format**

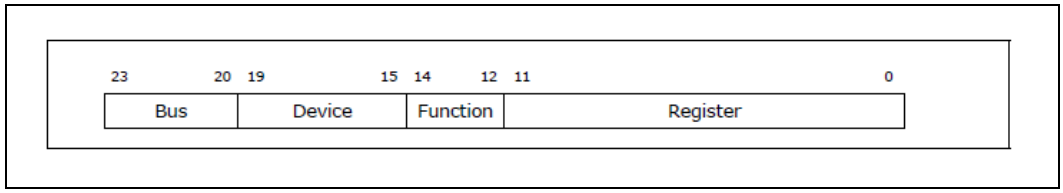



Table 30-21, "RdPCIConfigLocal() PECI Proxy Block Write" shows the RdPCIConfigLocal PECI proxy block write format and Table 30-22, "RdPCIConfigLocal() PECI Proxy Block Read" shows the RdPCIConfigLocal PECI proxy block read format. It is not possible to access PCI Express* B0, D0, F0 using the RdPCIConfigLocal command.

### Table 30-21. RdPCIConfigLocal() PECI Proxy Block Write

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode Command Code | | 0x62 | | |
| Byte Count | N | Control Data | 0x09 | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | 0x05 | | |
| Byte 4 | PECI Read Length | | {0x02,0x3, 0x5} | | One byte for the Completion code plus byte count of the desired data return from the accessed configuration-space register:<br>• 0x02 = One byte desired.<br>• 0x03 = One word desired.<br>• 0x05 = One DWord desired. |
| Byte 5 | RdPCIConfigLocal() Command Code | | 0xe1 | | |
| Byte 6 | Host ID and Retry | | 0x00 | | |
| Byte 7 | PCI Configuration Address Byte 1 (LSB) | | 24-bit PCI Configuration Address | | 24-bit PCI configuration address mapping:<br>• Bit [11:0] = Register<br>• Bit [14:12] = Function<br>• Bit [19:15] = Device<br>• Bit [23:20] = Bus |
| Byte 8 | PCI Configuration Address Byte 2 | | | | |
| Byte 9 | PCI Configuration Address Byte 3 (MSB) | | | | |

## Table 30-22. RdPCIConfigLocal() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x04, 0x05, 0x07} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error code defined in Table 30-6, "PECI Proxy Read."<br>N = {0x4,0x5,0x7} indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte [7:3] = PECI response data<br>N = {0x04, 0x05, 0x07} value is determined by the number of PECI response data bytes returned in Byte [7:4] on the SMBus.<br>N Value:<br>• 0x04 = 1 data byte<br>• 0x05 = 2 data bytes<br>• 0x07 = 4 data bytes |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte [1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte [2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |
| Byte 4 | PCI Data 1 (LSB) | | Data = 1,2 or 4 bytes | | Data returned from the PCI configuration space. Data is either a byte, word, or DWord depending on the PECI read length field of the associated PECI proxy block write. |
| Byte 5 | PCI Data 2 | | | | |
| Byte 6 | PCI Data 3 | | | | |
| Byte 7 | PCI Data 4 (MSB) | | | | |

## 30.7.8 WrPCILocalConfig()

The WrPCILocalConfig() command provides sideband write access to the PCI configuration space that resides within the processor. PECI originators can access this space even before BIOS enumeration of the system buses. The write accesses to registers that are locked will not take effect but will still return a completion code of 0x40. However, write accesses to registers that are hidden will return a completion code of 0x90.

Table 30-23, "WrPCILocalConfig() PECI Proxy Block Write" shows the WrPCILocalConfig PECI proxy block write format and Table 30-24, "WrPCILocalConfig() PECI Proxy Block Read" shows the WrPCILocalConfig PECI proxy block read format.

**Table 30-23. WrPCILocalConfig() PECI Proxy Block Write (Sheet 1 of 2)**

| SMBUS | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBUS Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | {0x0b, 0x0c, 0x0e} | | |
| Byte 1 | SMBUS-PECI Handshake Control | | 0x01 | | Value = 0x01<br><br>AW FCS required<br><br>Bridge Controller generates the AW FCS and appends it to the PECI command to the client. |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | {0x07, 0x08, 0x0a} | | Refer to later Section. |
| Byte 4 | PECI Read Length | | 0x01 | | |
| Byte 5 | WrPCILocalConfig () Command Code | | 0xe5 | | |
| Byte 6 | Host ID & Retry | | 0x00 | | |

### Table 30-23. WrPCILocalConfig() PECI Proxy Block Write (Sheet 2 of 2)

| SMBUS | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Byte 7 | PCI Configuration Address Byte 1 (LSB) | PECI Command | 24-bit PCI Configuration Address | BMC | PCI Configuration Address Byte[3:1] = 24-Bit PCI Configuration Address<br>Mapping:<br>• Bit[11:0] = Register<br>• Bit[14:12] = Function<br>• Bit[19:15] = Device<br>• Bit[23:20] = Bus<br>See supported Structure sections for details. |
| Byte 8 | PCI Configuration Address Byte 2 | | | | |
| Byte 9 | PCI Configuration Address Byte 3 (MSB) | | | | |
| Byte 10 | PECI Data 1 (LSB) | | Data = (1, 2, or 4 bytes) | | |
| Byte 11 | PECI Data 2 | | | | |
| Byte 12 | PECI Data 3 | | | | |
| Byte 13 | PECI Data 4 (MSB) | | | | |
| Byte 14 | AW FCS | | 0x00 | | PECI Command Controller generates the AW FCS based on the setting of Bit[0] in Byte 1. |

### Table 30-24. WrPCILocalConfig() PECI Proxy Block Read

| SMBUS | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBUS Command | 0x4B | BMC | |
| SMBUS Read Command Code | SMBUS Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x03} | PECI "Command Controller" | N = 0x01 for Busy state<br>• Byte 1 = 0x01<br>N= 0x02 for Transaction Errors<br>• Byte 1 = 0x02<br>• Byte 2 =Error Code defined Table 30-6, "PECI Proxy Read.".<br>N={0x03} indicates successful PECI Transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte 3 = PECI Completion Code |
| Byte 1 | Status | | CMD_STAT | | See Byte[1] of Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte[2] of Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion Code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |

## 30.7.9 RdEndPointConfig()

The RdEndPointConfig() command provides sideband read access to the PCI configuration space that resides within the processor, as well as the SSA sideband configuration space of each agent. The exact listing of supported devices, functions, and registers is outside the scope of this document. PECI originators can access this space even before BIOS enumeration of the system busses. PECI originators may also conduct a device/function/register enumeration sweep of this space by issuing reads in the same manner that the BIOS would.

Table 30-25, "RdEndPointConfig() PECI Proxy Block Write" shows the RdEndPointConfig PECI proxy block write format and Table 30-26, "RdEndPointConfig() PECI Proxy Block Read" shows the RdEndPointConfig PECI proxy block read format.

**Table 30-25. RdEndPointConfig() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x0B | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | {0x07} | | |
| Byte 4 | PECI Read Length | | {0x02, 0x3, 0x5} | | |
| Byte 5 | RdEndPointConfig() Command Code | | 0xC1 | | Use the same command for both, i.e., RdEndPointConfig is an alias. |
| Byte 6 | Host ID and Retry | | 0x00 | | |
| Byte 7 | SoC Sideband Port | | | | Endpoint Address |
| Byte 8 | Port Register Address Byte 1 (LSB) | | Any valid register address for the sideband port | | Port offset<br>If the addressed sideband port register does not exist, an error is returned. |
| Byte 9 | Port Register Address Byte 2 | | | | |
| Byte 10 | Port Register Address Byte 3 | | | | |
| Byte 11 | Port Register Address Byte 4 (MSB) | | | | |

## Table 30-26. RdEndPointConfig() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x1, 0x02, 0x04, 0x05, 0x07} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error code defined in Table 30-6, "PECI Proxy Read."<br>N = {0x4,0x5,0x7} indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte [7:3] = PECI response data<br>N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte [7:4] on the SMBus.<br>N Value:<br>• 0x04 = 1 data byte<br>• 0x05 = 2 data bytes<br>• 0x07 = 4 data bytes |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte[2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion Code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |
| Byte 4 | Data 1 (LSB) | | Data = 1, 2, or 4 bytes | | |
| Byte 5 | Data 2 | | | | |
| Byte 6 | Data 3 | | | | |
| Byte 7 | Data 4 (MSB) | | | | |

## 30.7.10 WrEndPointConfig()

The WrEndPointConfig() command provides sideband write access to the PCI configuration space that resides within the processor. PECI originators can access this space even before BIOS enumeration of the system busses. The exact listing of supported devices, functions, and registers for writing is outside the scope of this document. Refer to the appropriate processor specifications for details on registers that are accessible through this command.

Table 30-27, "WrEndPointConfig() PECI Proxy Block Write" shows the RdEndPointConfig PECI proxy block write format and Table 30-28, "WrEndPointConfig() PECI Proxy Block Read" shows the RdEndPointConfig PECI proxy block read format.

**Table 30-27. WrEndPointConfig() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode Command Code | | 0x62 | | |
| Byte Count | N | Control Data | {0x0D, 0x0E, 0x10} | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x01 | | Always set the value of Byte 1 to 0x01. |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | |
| Byte 3 | PECI Write Length | | {0x09, 0x0A, 0x0C} | | |
| Byte 4 | PECI Read Length | | 0x01 | | |
| Byte 5 | WrEndPointConfig() Command Code | | 0xC5 | | |
| Byte 6 | Host ID & Retry | | 0x00 | | |
| Byte 7 | SoC Sideband Port | | 0x04 | | |
| Byte 8 | Register Byte 1 (LSB) | | Address | | |
| Byte 9 | Register Byte 2 | | | | |
| Byte 10 | Register Byte 3 | | | | |
| Byte 11 | Register Byte 4 (MSB) | | | | |
| Byte 12 | PECI Data 1 (LSB) | | Data = 1, 2, or 4 bytes | | Sidebaoperation. |
| Byte 13 | PECI Data 2 | | | | |
| Byte 14 | PECI Data 3 | | | | |
| Byte 15 | PECI Data 4 (MSB) | | | | |
| Byte 16 | AW FCS | | 0x00 | | Always set the value of Byte 16 to 0x00. |

### Table 30-28. WrEndPointConfig() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x03} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for Transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error code defined in Table 30-6, "PECI Proxy Read."<br>N = {0x03} indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte3 = PECI completion code |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte[1] defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte[2] defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |

## 30.7.11 MCA Access via RdEndPointConfig()

An external agent can access the MCA banks over sideband using the RdEndPointConfig() command.

The registers that are accessed are shown in Table 30-29, "MCA Access via RdEndPointConfig()."

**Table 30-29. MCA Access via RdEndPointConfig()**

| Register Type | Location | Register | MSR Address |
|---|---|---|---|
| MC Bank 4 | Bunit | IA32_MC4_CTL | 410h |
| | | IA32_MC4_STATUS | 411h |
| | | IA32_MC4_ADDR | 412h |
| | | IA32_MC4_MISC | 413h |
| MC Bank 5 | P-Unit (PMU) | IA32_MC5_CTL | 414h |
| | | IA32_MC5_STATUS | 415h |
| MC Bank 6 | Aunit | IA32_MC6_CTL | 418h |
| | | IA32_MC6_STATUS | 419h |
| | | IA32_MC6_ADDR | 41Ah |
| MC Bank 7 | Dunit0 | IA32_MC7_CTL | 41Ch |
| | | IA32_MC7_STATUS | 41Dh |
| | | IA32_MC7_MISC | 41Fh |
| MC Bank 8 | Dunit1 | IA32_MC8_CTL | 420h |
| | | IA32_MC8_STATUS | 421h |
| | | IA32_MC8_MISC | 423h |

## 30.7.12 Endpoints supporting Rd/WrPCILocalConfig

The following endpoints can be accessed by the RdPCILocalConfig command.

**Table 30-30. Endpoints Supporting Rd/WrPCIConfigLocal**

| IP Block | PCI B:D:F | CfgRd |
|---|---|---|
| VRP2 (for QAT) | B0:D6:F0 | x |
| PCIe RP[0] | B0:D9:F0 | X |
| PCIe RP[1] | B0:D10:F0 | X |
| PCIe RP[2] | B0:D11:F0 | X |
| PCIe RP[3] | B0:D12:F0 | X |
| PCIe RP[4] | B0:D14:F0 | X |
| PCIe RP[5] | B0:D15:F0 | X |
| PCIe RP[6] | B0:D16:F0 | X |
| PCIe RP[7] | B0:D17:F0 | X |
| SATA Ctrl0 | B0:D19:F0 | X |
| SATA Ctrl1 | B0:D20:F0 | X |
| USB3 | B0:D21:F0 | X |
| VRP0 (for LAN Ctrl0) | B0:D22:F0 | X |
| VRP1 (for LAN Ctrl1) | B0:D23:F0 | X |
| HSUART | B0:D26:F{0, 1, 2} | X |
| LANCtrl0 | By:Dx:F{0, 1, 2, …, 63} | X |
| LANCtrl1 | Bz:Dx:F{0, 1, 2, …, 63} | x |

## 30.7.13 Endpoints supporting Rd/WrEndPointConfig

The following endpoints can be accessed by the Rd/WrEndPointConfig command

**Table 30-31. Endpoints Supporting Rd/WrEndPointConfig**

| EndPoint | Port ID (hex) | CfgRd | CfgWr |
|---|---|---|---|
| PUNIT0 | 46h | x | X |
| DUNIT_Shared | 16h | X | |
| DUNIT0 | 10h | X | |
| DUNIT1 | 12h | X | |
| BUNIT0 | 4Ch | X | |
| TUNIT0 | 50h | X | |
| USB210 | A7h | X | |
| PCIE1.D14 | B3h | X | |
| PCIE1.D15 | 8Ah | X | |
| PCIE1.D16 | 8Bh | X | |
| PCIE1.D17 | 8Ch | X | |
| PCIE0.D9 | B4h | X | |
| PCIE0.D10 | 8Dh | X | |
| PCIE0.D11 | 8Eh | X | |
| PCIE0.D12 | 8Fh | X | |
| SATA0:D19 | B5h | X | |
| SATA1.D20 | B6h | X | |
| KRM0 | 76h | X | |
| MAC0 | 77h | X | |
| MAC1 | 78h | X | |

## 30.8 DRAM Thermal Capabilities

Various DRAM component temperature data can be accessed using the RdPkgConfig() and WrPkgConfig() PECI commands addressed to the SoC.

See Section 30.7.4, "RdPkgConfig()" and Section 30.7.5, "WrPkgConfig()" for the format of the commands. The accessible data structures are given in Table 30-32, "Summary of DRAM Thermal and Power Optimization Services" which contains the values of the command index, parameter, and PECI data fields.

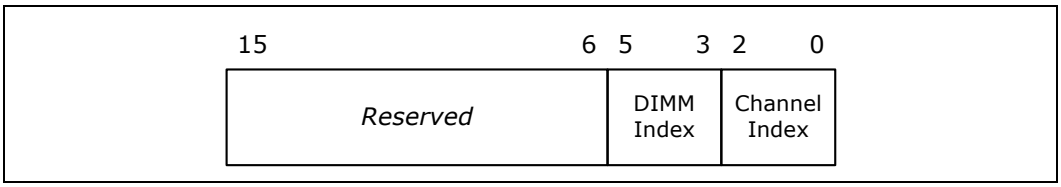
**Table 30-32. Summary of DRAM Thermal and Power Optimization Services (Sheet 1 of 2)**

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Accumulated DRAM Energy Read | 04 | 0xFF | DRAM Energy consumed by the DIMMs | Read the DRAM energy consumed by all the DIMMS in all the channels | DDR_ENERGY_STATUS (CSR), MSR_DRAM_ENERGY_STATUS (MSR 619h) |
| RdPkgConfig()<br><br>DIMM Temperature Reading (CLTT only) | 14 | Channel Index | Absolute temperature in degrees Celsius: [7:0] DIMM#0<br>[15:8] DIMM#1<br>[23:16] DIMM#2<br>[31:24] DIMM#3<br>For OLTT, will return 0 | Read the temperature of each DIMM within a channel | C[0/1]TEMPSTS[0/1] |
| WrPkgConfig()<br><br>DRAM Rank Temperature Write | 18 | Channel Index and DIMM Index | Absolute temperature in degrees Celsius for ranks 0-3 | Write the temperature for each rank within a single DIMM | None |
| RdPkgConfig()<br><br>DRAM Channel Temperature Read (CLTT only) | 22 | 0x0000 | Absolute temperatures in degrees Celsius<br>[7:0] Channel#0<br>[15:8] Channel#2<br>[31:16] Reserved<br>For OLTT, will return 0 | Read the maximum DRAM channel temperature | None |
| WrPkgConfig()<br>RdPkgConfig()<br><br>DRAM Power Limit Data | 34 | 0x0000 | DRAM Plane Power Limit Data | Write/Read DRAM Power Limit Data | DDR_RAPL_LIMIT (CSR), MSR_DRAM_POWER_LIMIT (MSR 618h) |

### Table 30-32. Summary of DRAM Thermal and Power Optimization Services (Sheet 2 of 2)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>DRAM Power Info Read | 35 | 0x0000 | Maximum DRAM power settings & maximum time window | Read DRAM power settings info to be used by power limiting entity | DRAM_POWER_INFO[63:32] (CSR) , MSR_DRAM_POWER_INFO[63:32] (MSR 61Ch) |
| RdPkgConfig()<br><br>DRAM Power Info Read | 36 | 0x0000 | Typical and minimum DRAM power settings | Read DRAM power settings info to be used by power limiting entity | DRAM_POWER_INFO[31:0] (CSR) , MSR_DRAM_POWER_INFO[31:0] (MSR 61Ch) |
| RdPkgConfig()<br><br>DRAM Power Limit Performance Status Read | 38 | 0x0000 | Accumulated DRAM throttle time | Read sum of all time durations for which each DRAM has been throttled | DDR_RAPL_PERF_STATUS (CSR), MSR_DRAM_PERF_STATUS (MSR 61Bh) |

### Figure 30-9. Channel Index and DIMM Index Parameter Word



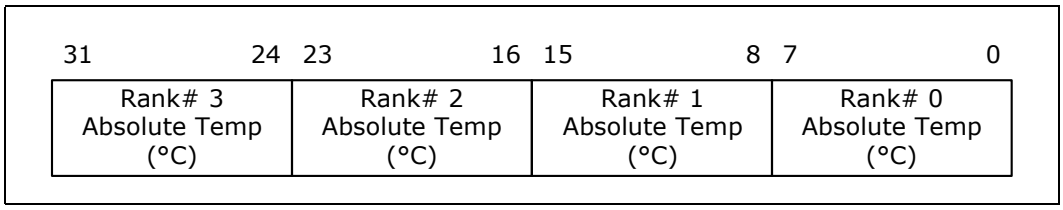

### Table 30-33. Channel Index and DIMM Index

| Index Encoding | Physical Channel Number | Physical DIMM Number |
|---|---|---|
| 000 | 0 | 0 |
| 001 | 1 | 1 |
| 010 | Reserved | Reserved |
| 011 | Reserved | Reserved |

### 30.8.1 DRAM Rank Temperature Write (Index = 18)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." The DRAM rank temperature write allows the PECI host to program the processor with the temperature for all the ranks within a DIMM up to a maximum of four ranks. The programming data structure is defined in Figure 30-10, "Write DRAM Rank Temperature Data Dword."

**Figure 30-10.Write DRAM Rank Temperature Data Dword**

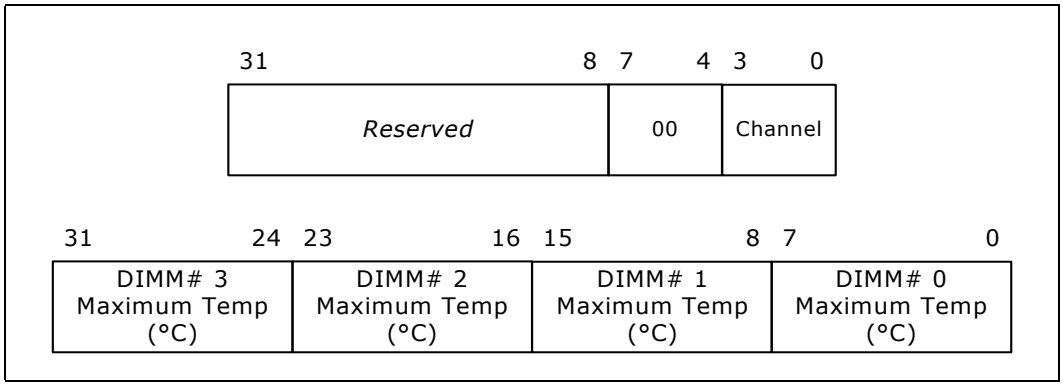



The DIMM index and channel index are specified through the parameter field as shown in Figure 30-9, "Channel Index and DIMM Index Parameter Word" and Table 30-33, "Channel Index and DIMM Index." This write is relevant to platforms that do not have on-die or on-board DIMM thermal sensors to provide memory temperature information or if the processor does not have direct access to the DIMM thermal sensors. This temperature information is used by the processor in conjunction with the activity-based DRAM temperature estimations. With this SoC product family, while temperature values are accepted on a per-rank basis, the SoC uses the average temperature over a given channel for throttling decisions.

### 30.8.2 DIMM Temperature Read (Index = 14)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This feature allows the PECI host to read the maximum temperature of all the DIMMs within a channel. If CLTT is disabled, the values returned will be 0. The SoC does not support activity-based temperature estimations. The returned data structure is shown in Figure 30-11, "Read DIMM Temperature Data Dword."

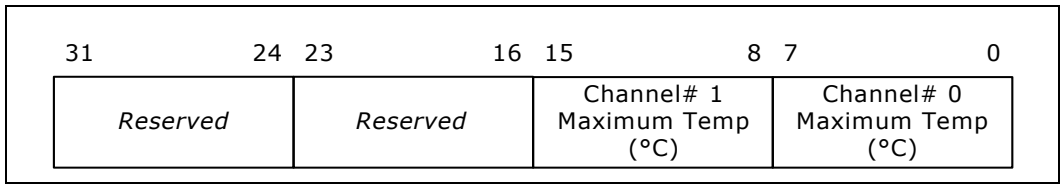
**Figure 30-11.Read DIMM Temperature Data Dword**



## 30.8.3 DRAM Channel Temperature Read (Index = 22)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This feature enables a PECI host read of the maximum temperature of each memory controller channel of the given product SKU. This includes all of the DIMMs within the channel and all the ranks within each of the DIMMs. If CLTT is disabled, the values returned will be 0 See the returned data structure as shown in Figure 30-12, "Read DRAM Channel Temperature Data Dword."

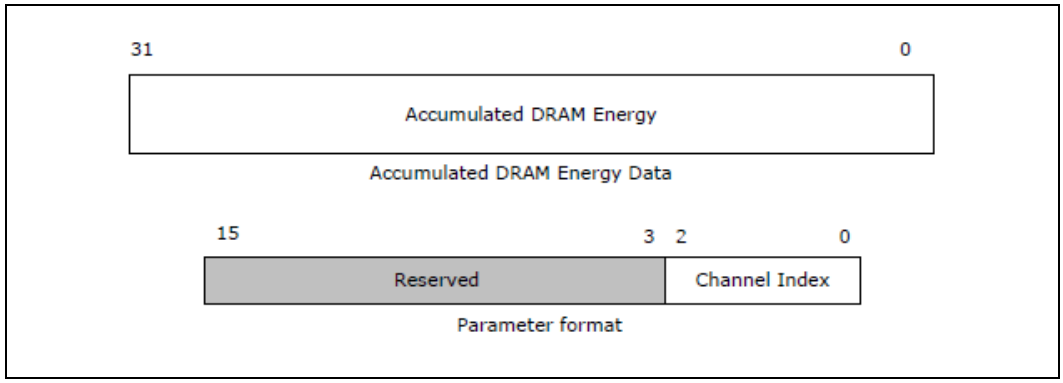
**Figure 30-12.Read DRAM Channel Temperature Data Dword**




## 30.8.4 DRAM Accumulated Energy Read (Index = 4)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This feature allows the PECI host to read the DRAM energy consumed by all the DIMMs in all the channels or all the DIMMs within just a specified channel. The channel index are specified through the parameter field. Units used are defined as per the Package Power SKU Unit read described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)." The data is tracked by a 32-bit counter that wraps around. The channel index in Figure 30-14, "Read DRAM Power Info Data Dword" is specified as per the index encoding described in Figure 30-25, "Temperature Target Read." While Intel requires reading the accumulated energy data at least once every 16 seconds to ensure functional correctness, a more realistic polling rate recommendation is once every 100 mS for better accuracy. This feature assumes a 200W memory capacity. In general, as the power capability decreases, so will the minimum polling rate requirement.

**Figure 30-13.Read DRAM Accumulated Energy Data Dword**



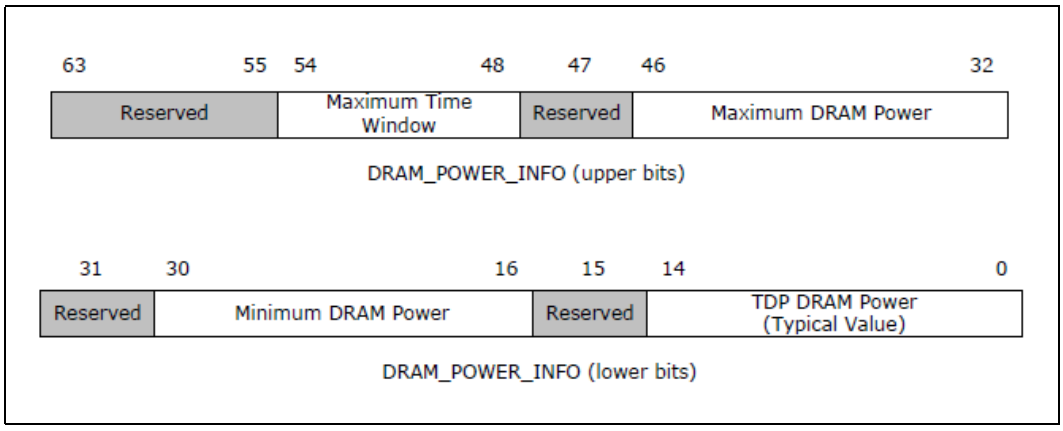
### 30.8.5 DRAM Power Info Read (Index = 35/36)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This read returns the minimum, typical and maximum DRAM power settings and the maximum time window over which the power can be sustained for the entire DRAM domain and is inclusive of all the DIMMs within all the memory channels. Any power values specified by the power limiting entity that is outside of the range specified through these settings cannot be guaranteed. Since this data is 64 bits wide, PECI facilitates access to this register by allowing two requests to read the lower 32 bits and upper 32 bits separately. Power and time units for this read are defined as per the Package Power SKU Unit settings described in Section 30.9, "CPU Thermal and Power Optimization Capabilities."

The minimum DRAM power corresponds to a minimum bandwidth setting of the memory interface. It does 'not' correspond to a processor IDLE or memory self-refresh state. The 'time window' in Figure 30-14, "Read DRAM Power Info Data Dword" is representative of the rate at which the SoC Power Management Unit (PMU) samples the DRAM energy consumption information and reactively takes the necessary measures to meet the imposed power limits. Programming too small a time window may not give the PMU enough time to sample energy information and enforce the limit while too large a time window runs the risk of the PMU not being able to monitor and take timely action on energy excursions. While the DRAM power setting in Figure 30-14, "Read DRAM Power Info Data Dword" provides a maximum value for the 'time window' (typically a few seconds), the minimum value may be assumed to be ~100 mS.

The Power Management Controller (PMC) programs the DRAM power settings described in Figure 30-14, "Read DRAM Power Info Data Dword" when DRAM characterization has been completed by the memory reference code (MRC) during boot as indicated by the setting of the MC_BIOS_REQ register. The DRAM power settings will be programmed during boot independent of the 'DRAM Power Limit Enable' bit setting. In general, any tuning of the power settings is done by polling the voltage regulators supplying the DIMMs.

**Figure 30-14.Read DRAM Power Info Data Dword**



## 30.8.6 DRAM Power Limit Data Read/Write (Index = 34)

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This feature allows the PECI host to program the power limit over a specified time or control window for the entire DRAM domain covering all the DIMMs within all the memory channels. Actual values are chosen based on DRAM power consumption characteristics. The units for the DRAM Power Limit and Control Time Window are determined as per the Package Power SKU Unit settings described in Table 30.9.2, "Package Power SKU Unit Read (Index = 30)." The DRAM Power Limit Enable bit should be set to activate this feature. Exact DRAM power limit values are largely determined by platform memory configuration. As such, this feature is disabled by default and there are no defaults associated with the DRAM power limit values. The PECI host may be used to enable and initialize the power limit fields for the purposes of DRAM power budgeting. Alternatively, this can also be accomplished through inband writes to the appropriate registers. Both power limit enabling and initialization of power limit values can be done in the same command cycle. All RAPL parameter values including the power limit value, control time window, and enable bit will have to be specified correctly even if the intent is to change just one parameter value when programming over PECI.

The following conversion formula should be used for encoding or programming the 'Control Time Window' in bits [23:17].

Control Time Window (in seconds) = ([1 + 0.25 * 'x'] * $2^{'y'}$) * 'z' where

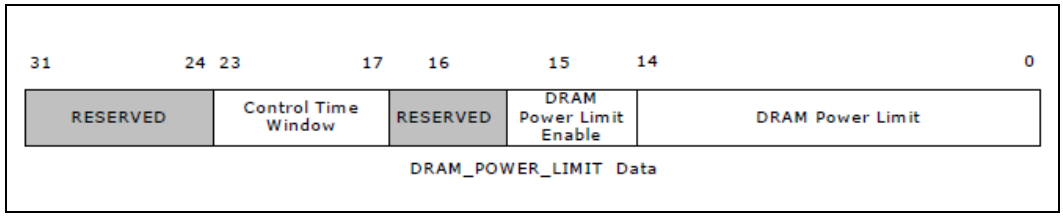
'x' = integer value of bits[23:22]

'y' = integer value of bits[21:17]

'z' = Package Power SKU Time Unit[19:16] (see Table 30.9.2, "Package Power SKU Unit Read (Index = 30)" for details on Package Power SKU Unit)

For example, using this formula, a control time value of 0x0A will correspond to a '1-second' time window.

**Figure 30-15.DRAM Power Limit Data**

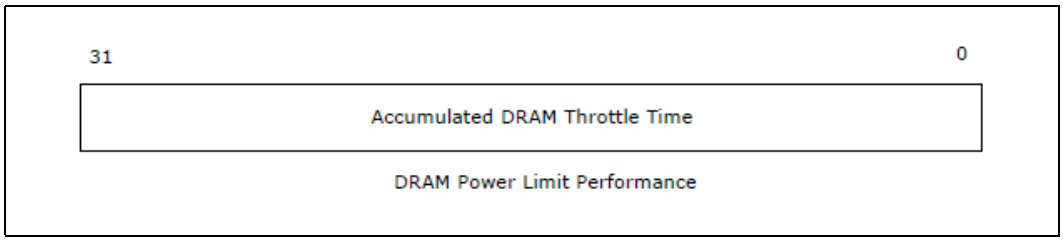



From a DRAM power management standpoint, all post-boot DRAM power management activities (also referred to as 'DRAM RAPL' or 'DRAM Running Average Power Limit') should be managed exclusively through a single interface like PECI or alternatively an inband mechanism. If PECI is being used to manage DRAM power budgeting activities, BIOS should lock out all subsequent inband DRAM power limiting accesses by setting bit 31 of the DRAM_POWER_LIMIT MSR or DRAM_PLANE_POWER_LIMIT CSR to '1'.

## 30.8.7 DRAM Power Limit Performance Status Read

See Table 30-32, "Summary of DRAM Thermal and Power Optimization Services." This feature allows the PECI host to assess the performance impact of the currently active DRAM power limiting modes. The read return data contains the sum of all the time durations for which each of the DIMMs has been operating in a low power state. This information is tracked by a 32-bit counter that wraps around. The unit for time is determined as per the Package Power SKU Unit settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)" The DRAM performance data does not account for stalls on the memory interface.

**Figure 30-16.DRAM Power Limit Performance Data**




In general, for the purposes of DRAM RAPL, the DRAM power management entity should use PECI accesses to DRAM energy and performance status in conjunction with the power limiting feature to budget power between the various memory sub-systems in the server system.

## 30.9 CPU Thermal and Power Optimization Capabilities

Table 30-34, "Summary of CPU Thermal and Power Optimization Services" provides a summary of the power and thermal optimization capabilities that can be accessed over PECI for this SoC product family. Twenty-three services are shown and each is described in the subsections following the table.

*Note:* The index field values are referenced as decimal numbers.

Table 30-34, "Summary of CPU Thermal and Power Optimization Services" also shows alternate in-band mechanisms to access similar or equivalent information where applicable for register read and write services.

The BIOS is required to populate CPUID, PlatformID and the CPU microcode update revision.

**Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 1 of 7)**

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Package Identifier Read<br>Section 30.9.1, on page 848 | 00 | 0x0000 | CPU ID Information | Returns processor-specific information including CPU family, model and stepping information. | Execute the CPUID instruction to get the processor signature. |
| | | 0x0001 | Platform ID | Used to ensure microcode update compatibility with the processor. | IA32_PLATFORM_ID (MSR 17h) |
| | | 0x0002 | PCU Device ID | Returns the device ID information for the processor PMU. | |
| | | 0x0003 | Max Thread ID | Returns the maximum thread ID value supported by the processor. | |
| | | 0x0004 | CPU Microcode Update Revision | Returns the processor microcode and internal power control-unit firmware revision information. | IA32_BIOS_SIGN_ID (MSR 8Bh) |
| | | 0x0005 | MCA Error Source Log | Returns the MCA error source log. | MCA_ERR_SRC_LOG (CSR) |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 2 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| WrPkgConfig()<br><br>VR Debug Interface | 01 | 0x0000 | [03:00] VR ID.<br>00h Vcc<br>01h Vnn<br>03h VccSram<br>02h Memory<br>06h [08:04] VR command. GetReg (0x07) supported. For illegal commands, return ILLEGAL_CMD completion code.<br>[15:09] Reserved. Fail with ILLEGAL_DATA if non-zero.<br>[23:16] VR register address or VID data.<br>[31:24] Reserved. Fail with ILLEGAL_DATA if non-zero. | Read VR Register data through the VR12.x GetReg command. Data is returned in WrPkgConfig() response | |
| RdPkgConfig()<br><br>Package Temperature Read<br>Section 30.9.5, "Package Temperature Read (Index = 2)" | 02 | | [15:0] Reserved<br>[22:16] Processor package Temperature relative to PROCHOT | Returns the maximum processor die temperature (average) in PECI format. | P_CR_PACKAGE_THERM_STATUS (CSR) |
| RdPkgConfig()<br><br>Accumulated Energy Status Read<br>Section 30.9.11, "Accumulated Energy Status Read (Index = 3)" | 03 | 0x0 = Vcc Energy<br>0xFF = SoC (Total Pkg Energy) | Accumulated CPU energy | Returns the value of the energy consumed by just the VCC power plane or the entire SoC. | PACKAGE_ENERGY_STATUS (CSR), MSR_PKG_ENERGY_STATUS (MSR 611h) |
| RdPkgConfig()<br><br>Efficient Performance Indication (EPI) | 06 | 0x0000 | Number of productive processor cycles | Read number of productive cycles for power budgeting purpose | None |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 3 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Package Power Limit Performance Status Read<br>Section 30.9.13, "Package Power Limit Performance Status Read (Index = 8)" | 08 | 0x00FF: CPU package | Accumulated CPU throttle time | Read the total time for which the processor package was throttled due to power limiting. | PACKAGE_RAPL_PERF_STATUS (CSR), MSR_PKG_PERF_STATUS (MSR 613h) |
| RdPkgConfig()<br><br>Per Core DTS Temperature Read<br>Section 30.9.6, "Per Core DTS Temperature Read (Index = 9)" | 09 | 0x0000 *through* 0x0017: Cores 0 *through* 15<br>**0xfd**: SoC temperature sensor 0<br>**0xfe**: SoC temperature sensor 1<br>**0xff**: SoC temperature sensor 2 | Per core DTS maximum temperature [15:0] Module MAX temperature (same format as in Package Temperature) [32:16] Reserved | Read the maximum DTS temperature of a particular core or the system agent within the processor die in the relative PECI temperature format. | IA32_THERM_STATUS (MSR 19C) |
| THERM_MARGIN | 10 | 0x10 | N/A | Format S16.7.8 Read the thermal margin to spec (DTS 2.0) Margin offset relative to T-Control. | P_CR_PACKAGE_THERM_MARGIN_0_0_0_MCHBAR (CSR)<br>PACKAGE_THERM_MARGIN (MSR 1A1h) |
| RdPkgConfig()<br><br>Socket Power Throttled Duration | 11 | 0x0000 | Duration of the SoC operating condition | Returns the duration for which RAPL power limit determined the operating condition | None |
| WrPkgConfig()<br><br>Configure Current Limit | 15 | 0x0000 | Current Limit | Configure the maximum current to limit the SoC power | MSR_VR_CURRENT_CONFIG (MSR 601h) |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 4 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Temperature Target Read<br>Section 30.9.7, "Temperature Target Read (Index = 16)" | 16 | 0x0000 | [7:0] Reserved<br>[15:8] TCONTROL value which indicates the relative offset from PROCHOT<br>[23:16] PROCHOT temperature where the Thermal Monitor is activated.<br>[27:24] Temperature offset in degrees (C) from the PROCHOT. If offset is set the Thermal Monitor will occur at lower than PROCHOT. | Returns the maximum processor junction temperature, PROCHOT and processor $T_{CONTROL}$. | TEMPERATURE TARGET (CSR), MSR_TEMPERATURE_TARGET (MSR 1A2h) |
| RdPkgConfig()<br><br>Current Limit Read<br>Section 30.9.10, "Current Limit Read (Index = 17)" | 17 | 0x00FF | Current Limit | Reads the current limit on the VCC power plane | MSR_VR_CURRENT_CONFIG[31:0] (MSR 601h) |
| RdPkgConfig()<br><br>Package Thermal Status | 20 | 0x0000 | Thermal Status Register | Read the thermal status register and optionally clear any log bits.The register includes status and log bits for TCC activation, PROCHOT_N assertion & critical temperature | IA32_PACKAGE_THERM_STATUS (MSR 1B1H) |
| WrPkgConfig()<br><br>Package Power limits For multiple Turbo Modes<br>Section 30.9.12, "Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)" | 26 | 0x0000 | Power Limit 1 Data | Write power limit 1 data in multiple turbo mode. | PACKAGE_RAPL_LIMIT (CSR), MSR_PKG_POWER_LIMIT (MSR 610h) |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 5 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Package Power Limits For multiple Turbo Modes<br>Section 30.9.12, "Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)" | 27 | 0x0000 | Power Limit 2 Data | Read power limit 2 data in multiple turbo mode. | PACKAGE_RAPL_LIMIT (CSR), MSR_PKG_POWER_LIMIT (MSR 610h) |
| RdPkgConfig()<br><br>Package Power SKU Read<br>Section 30.9.3, "Package Power SKU Read (Index = 28 and 29)" | 28 | 0x0000 | Package Power SKU [31:0] | Returns Thermal Design Power (TDP) and minimum package power for the SoC product SKU. | PACKAGE_POWER_SKU (CSR), MSR_PKG_POWER_INFO (MSR 614h) |
| RdPkgConfig()<br><br>Package Power SKU Read<br>Section 30.9.3, "Package Power SKU Read (Index = 28 and 29)" | 29 | 0x0000 | Package Power SKU[64:32] | Returns the maximum package power value for the SoC product SKU and the maximum time interval for which it can be sustained. | PACKAGE_POWER_SKU (CSR), MSR_PKG_POWER_INFO (MSR 614h) |
| RdPkgConfig()<br><br>Package Power SKU Unit Read<br>Section 30.9.2, "Package Power SKU Unit Read (Index = 30)" | 30 | 0x0000 | Time, Energy and Power Units | Read units for power, energy and time used in the power control registers. | PACKAGE_POWER_SKU_UNIT (CSR), MSR_RAPL_POWER_UNIT (MSR 606h) |
| RdPkgConfig()<br><br>Total Time<br>Section 30.9.4, "Accumulated Run Time Read (Index = 31)" | 31 | 0x0000 | Elapsed time | Returns the total run time in ms. Total accumulated run time reported in IO_TSC_LOW. 31:0 - lower 32 bits of the UNCORE Time Stamp Counter. | IA32_TIME_STAMP_COUNTER (MSR 10h) |
| | | 0x00FE | 64-bit counter indicating the aggregate of C0 residency count from all cores | | |
| | | 0x00FF | 32 MSBs of 64-bit counter indicating the aggregate of C0 residency count from all cores | | |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 6 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| RdPkgConfig()<br><br>Thermally Constrained Time Read<br>Section 30.9.9, "Thermally Constrained Time Read (Index = 32)" | 32 | 0x0000 | Thermally Constrained Time | Read the time for which the processor has been operating in a lowered power state due to internal TCC activation. | None |
| WrPkgConfig()<br>RdPkgConfig()<br><br>ACPI P-T Notify Write/Read | 33 | 0x0000 | New p-state equivalent of P1 used in conjunction with package power limiting | Notify/Read the processor PCU to determine the p-state that is one state below the turbo frequency as specified through the last ACPI Notify | None |
| RdPkgConfig()<br><br>Read Ratio constraints | 49 | | [0:0] fused or resolved<br>- 0h: fused<br>- 1h: resolved<br>[4:1] AVX level<br>- 0h: non- AVX<br>- 1h: AVX<br>- 2h: AVX3<br>[11:5] core count input (0h - 7Fh). | Input value for core count returns turbo ratio limits for core counts.<br>4x+1, 4x+2, 4x+3, 4x+4 in bits 0-7, 8-15, 16-23, 24-31<br>If 4x+y, y = {1,2,3,4} is not a valid core count, returns FFh as the limit. | |
| WrPkgConfig()<br>RdPkgConfig()<br><br>Read/Write Ratio Limit | 50 | 0x0000 | Ratio Limit | Reads/ Writes Turbo Ratio Limit | |

## Table 30-34. Summary of CPU Thermal and Power Optimization Services (Sheet 7 of 7)

| Service | Index Field (Decimal) | Parameter Field (Word) | PECI Data Field (DWord) | Description | Alternate In-band MSR or CSR Access |
|---|---|---|---|---|---|
| PL3 | 57 | 0x0000 | PECI Configuration of Package PL3 Limit | [14:0] power limit in 2^(POWER_UNIT) watts<br>[15:15] Power limit control enable<br>[16:16] Reserved<br>[23:17] Control Time Windows<br>[30:24] Duty Cycle in percentage. Max value of 100%<br>[31:31] Lock bit. If set, this command is locked out from future writes and all other data in this command is ignored | |
| GetCrashDumpDetails | 51 | Get Crash dump details | | Get the details of the Crash Dump. | |
| GetCrashDumpFrame | 52 | Get Frame | | Gets a particular frame in the specified crash dump | |

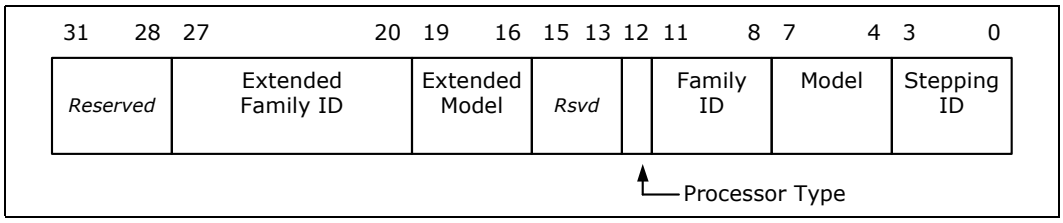
### 30.9.1 Package Identifier Read (Index = 0)

This feature enables the PECI host to uniquely identify the PECI client processor. The parameter field encodings shown in Table 30-34, "Summary of CPU Thermal and Power Optimization Services" allow the PECI host, typically the BMC, to access the relevant processor information as described below.

#### 30.9.1.1 CPU ID Information

This is data field contains the equivalent information that can be accessed through executing the Intel® architecture CPUID instruction. It contains the processor type, stepping, model, and family ID information as shown in Figure 30-17, "CPU ID Data."

**Figure 30-17.CPU ID Data**




#### 30.9.1.2 Platform ID

This data field can be used to ensure processor microcode updates are compatible with the processor.

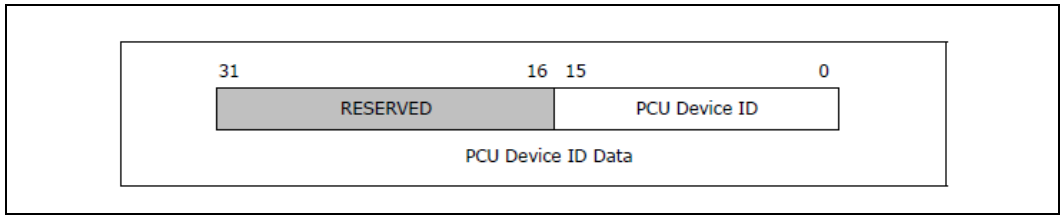
*Note:* The value of the platform ID or processor flag[2:0] as shown in Figure 30-18, "Platform ID Data" is typically unique to the platform type and processor stepping.

**Figure 30-18.Platform ID Data**

### 30.9.1.3 PCU Device ID

This information can be used to uniquely identify the processor power control unit (PCU) device when combined with the Vendor Identification register content and remains constant across all SKUs. Refer to the appropriate register description for the exact processor PCU Device ID value.

**Figure 30-19.PCU Device ID Data**

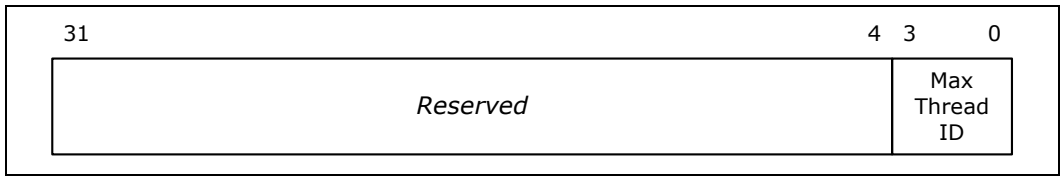



### 30.9.1.4 Maximum Thread ID

This is data field provides the number of supported processor threads.

*Note:* This value is dependent on the number of cores within the processor as determined by the processor SKU and is independent of whether certain cores or corresponding threads are enabled or disabled. See Figure 30-20, "Maximum Thread ID Data."

**Figure 30-20.Maximum Thread ID Data**

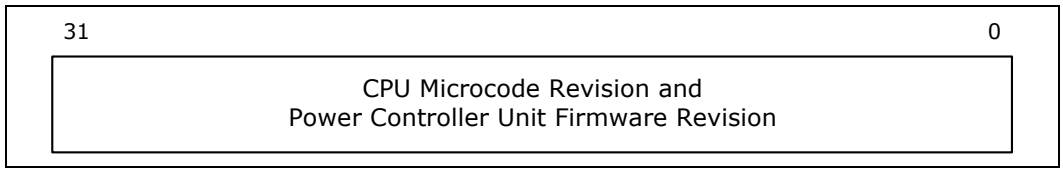



### 30.9.1.5 CPU Microcode Update Revision

This data field reflects the revision number for the microcode update and power control unit firmware updates on the processor sample.

*Note:* The revision data is a unique 32-bit identifier that reflects a combination of specific versions of the processor microcode and power control unit firmware. See Figure 30-21, "Processor Microcode Revision."

**Figure 30-21.Processor Microcode Revision**



#### 30.9.1.6 MCA Error Source Log

This is data field contains contents of the Machine-Check Architecture (MCA) Error Source Log register. See Figure 30-22, "Package Power SKU Unit Data" for details. The register indicates the value as defined when IERR and/or MCERR are indicated by the SoC.

### 30.9.2 Package Power SKU Unit Read (Index = 30)

This feature enables the PECI host to read the units of time, energy, and power used in the processor and DRAM power control registers for calculating power and timing parameters. In Figure 30-22, "Package Power SKU Unit Data," the default values are:

- Power unit field [3:0] = 0011b
- Energy unit field [12:8] = 10000b
- Time unit field [19:16] = 1010b

Actual unit values are calculated as shown in Table 30-35, "Power Control Register Unit Calculations."

**Figure 30-22. Package Power SKU Unit Data**

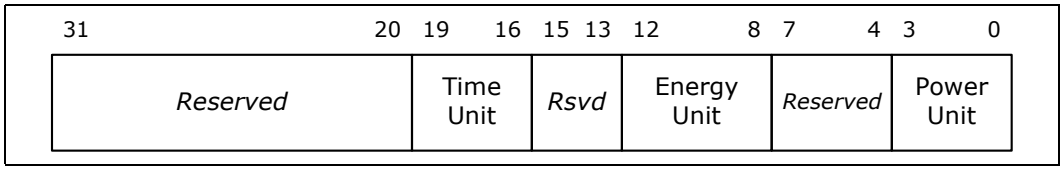


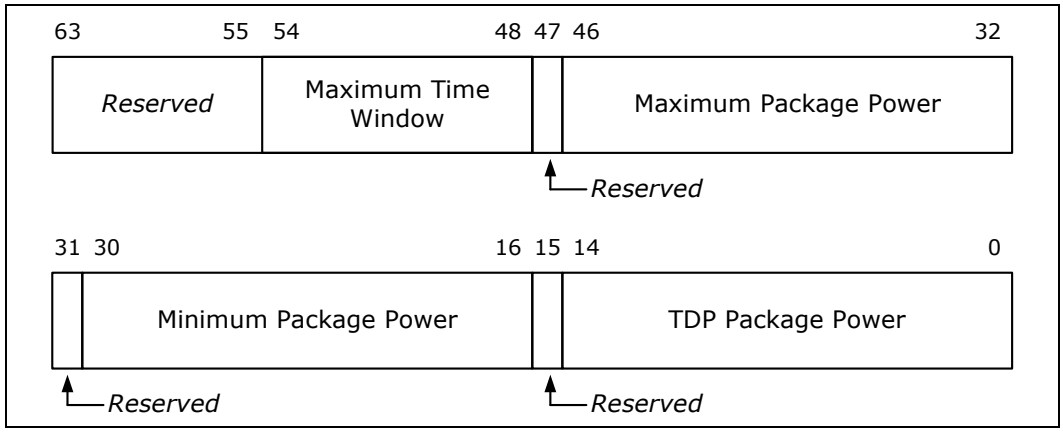

**Table 30-35. Power Control Register Unit Calculations**

| Unit Field | Value Calculation | Default Value |
|---|---|---|
| Time | 1s / 2^TIME UNIT | 1s / 2^10 = 976 µs |
| Energy | 1J/2^ENERGY UNIT | 1J/2^16 = 15.3 µJ |
| Power | 1W/2^POWER UNIT | 1W/2^3 = 1/8 W |

## 30.9.3 Package Power SKU Read (Index = 28 and 29)

This read allows the PECI host to access the minimum, Thermal Design Power (TDP) and maximum power settings for the processor package SKU. It also returns the maximum time interval or window over which the power can be sustained. If the power limiting entity specifies a power limit value outside of the range specified through these settings, power regulation cannot be guaranteed. Since this data is 64-bits wide, PECI facilitates access to this register by allowing two requests to read the lower 32 bits and upper 32 bits separately as shown in Figure 30-23, "Package Power SKU Data." Power units for this read are determined as per the package power SKU unit settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)."

**Figure 30-23.Package Power SKU Data**




The package power SKU data is programmed by the SoC internal Power Control Unit (PCU) firmware during boot time based on SKU dependent power-on default values set during SoC manufacturing. The TDP package power specified through bits [14:0] in Figure 30-23, "Package Power SKU Data" is the maximum value of the Power Limit1 field described in Section 30.9.12, "Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)" while the maximum package power in bits [46:32] is the maximum value of the Power Limit2 field which is also described in Section 30.9.12, "Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)."

The minimum package power in bits [30:16] is applicable to both the Power Limit1 and Power Limit2 fields and corresponds to a mode when all the cores are operational and in their lowest frequency mode. Attempts to program the power limit below the minimum power value may not be effective since the BIOS/OS, and not the integrated PCU, controls disabling of cores and core activity.

Maximum time window in bits [54:48] is representative of the maximum rate at which the internal PCU can sample the package energy consumption and reactively take the necessary measures to meet the imposed power limits. Programming too-large of a time window runs the risk of the PCU not being able to monitor and take timely action on package energy excursions. On the other hand, programming too-small of a time window may not give the PCU enough time to sample energy information and enforce the limit. The minimum value of the "time window" can be obtained by reading bits [21:15] of the PWR_LIMIT_MISC_INFO CSR using the PECI RdPCIConfigLocal() command.

### 30.9.4 Accumulated Run Time Read (Index = 31)

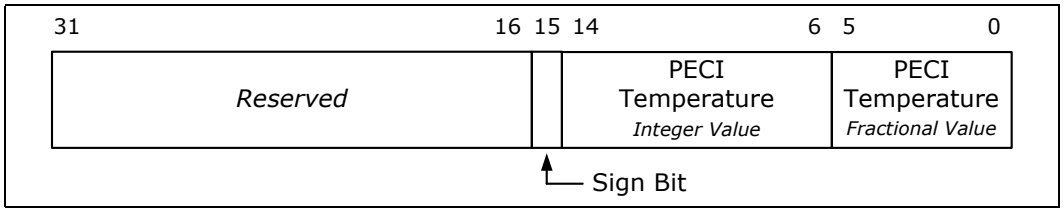
This services provides support for 2 accumulated runtime reads.

- 0x0000: This read returns the total time for which the processor has been executing with a resolution of 1 ms per count. This is tracked by a 32-bit counter that rolls over on reaching the maximum value. This counter activates and starts counting for the first time when the SoC desserts the active-low CPU_RESET_N output signal pin.
- 0x00FE: This read returns the total time for which the processor has been in C0 with a resolution of 1 ms per count. This is tracked by a 32-bit counter that rolls over on reaching the maximum value. This counter activates and starts counting for the first time when the SoC deasserts the active-low CPU_RESET_N output signal pin.

### 30.9.5 Package Temperature Read (Index = 2)

This read returns the maximum processor die temperature in a 16-bit PECI format. The upper 16 bits of the response data are reserved. See Figure 30-24, "Package Temperature Read Data." The PECI temperature data returned by this read is the instantaneous value and not the average value as is returned by the PECI GetTemp() described in Section 30.6.4, "PECI Proxy Command Trigger."

**Figure 30-24.Package Temperature Read Data**




### 30.9.6 Per Core DTS Temperature Read (Index = 9)

This feature enables the PECI host to read the maximum value of the Digital Thermal Sensor (DTS) temperature for any specific core within the processor. Alternatively, this service can be used to read the internal SoC System Agent (SSA) temperature. The temperature is returned in the same data format as described in Section 30.9.5, "Package Temperature Read (Index = 2)." Data is returned in a relative PECI temperature format.

Reads to a parameter value outside the supported range return an error as indicated by a completion code of 0x90. The supported range of parameter values can vary depending on the number of processor cores within the SoC. The temperature data returned through this feature is the instantaneous value and not the average value. It is updated once every 1 ms.

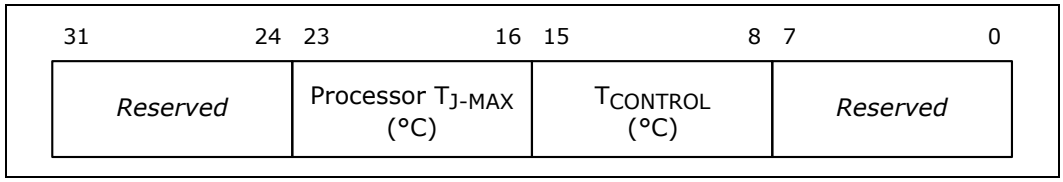
## 30.9.7 Temperature Target Read (Index = 16)

The temperature target read allows the PECI host to access the maximum processor junction temperature ($T_{J-MAX}$) in degrees Celsius. This is also the default temperature value at which the processor thermal control circuit activates. The $T_{J-MAX}$ value may vary from processor part to part to reflect manufacturing process variations.

The temperature target read also returns the processor $T_{CONTROL}$ value. The $T_{CONTROL}$ is returned in a standard PECI temperature format and represents the threshold temperature used by the thermal management system for a fan speed control.

See Figure 30-27.

**Figure 30-25.Temperature Target Read**




## 30.9.8 Package Thermal Status Read / Clear (Index = 20)

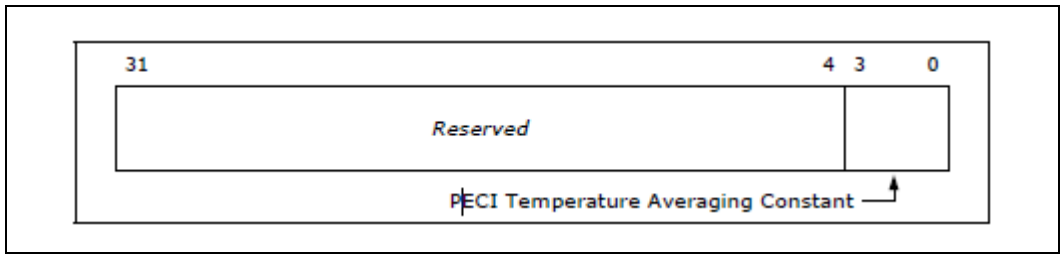
The Thermal Status Read provides information on package level thermal status. Data includes:

- Thermal Control Circuit (TCC) activation
- Bidirectional PROCHOT_N signal assertion
- Critical Temperature

Both status and sticky log bits are managed in this status word. All sticky log bits are set upon a rising edge of the associated status bit and the log bits are cleared only by Thermal Status reads or a processor reset. A read of the Thermal Status word always includes a log bit clear mask that allows the host to clear any or all of the log bits that it is interested in tracking.

A bit set to '0' in the log bit clear mask will result in clearing the associated log bit. If a mask bit is set to '0' and that bit is not a legal mask, a failing completion code will be returned. A bit set to '1' is ignored and results in no change to any sticky log bits. For example, to clear the TCC Activation Log bit and retain all other log bits, the Thermal Status Read should send a mask of 0xFFFFFFFD.

**Figure 30-26.Package Thermal Status Read/Clear**



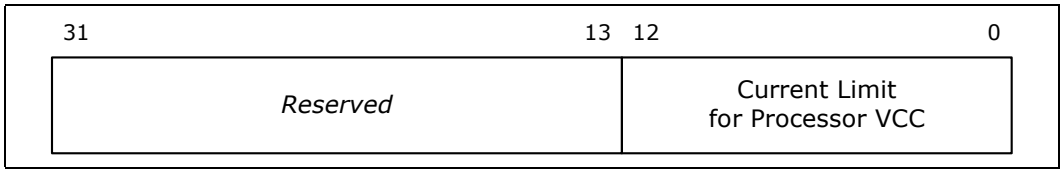
### 30.9.9 Thermally Constrained Time Read (Index = 32)

This feature allows the PECI host to access the total time for which the processor has been operating in a lowered power state due to Thermal Control Circuit (TCC) activation. The returned data includes the time required to ramp back up to the original P-State target after the TCC activation expires. This timer does not include TCC activation as a result of an external assertion of the SoC PROCHOT_N signal pin. This is tracked by a 32-bit counter with a resolution of 1 ms per count that rolls over or wraps around. Concerning the processor PECI clients, the only logic that can be thermally constrained are those which are supplied by VCC.

### 30.9.10 Current Limit Read (Index = 17)

This read returns the current limit for the processor VCC power plane in 1/8-ampere increments. Actual current limit data is contained only in the lower 13 bits of the response data. The default return value of 0x140 corresponds to a current limit value of 40 amperes. Bit 31 is a lock bit indicating that the IccMax has been locked to the returned value in bits 0-12.See Figure 30-27, "Current Limit Read Data."

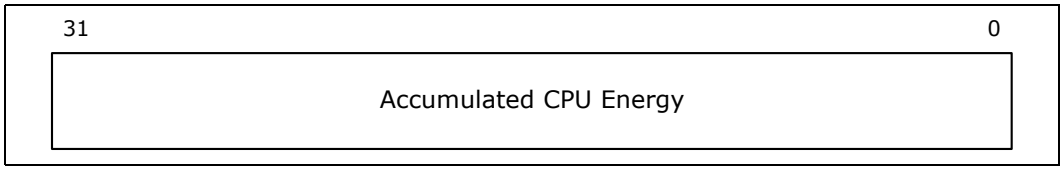
**Figure 30-27.Current Limit Read Data**




### 30.9.11 Accumulated Energy Status Read (Index = 3)

This service can return the value of the total energy consumed by the entire processor package or just the logic supplied by the VCC power plane as specified through the parameter field in Table 30-34, "Summary of CPU Thermal and Power Optimization Services." This information is tracked by a 32-bit counter that wraps around and continues counting on reaching its limit. See Figure 30-28, "Accumulated Energy Read Data." Energy units for this read are determined as per the package power SKU unit settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)."

**Figure 30-28.Accumulated Energy Read Data**

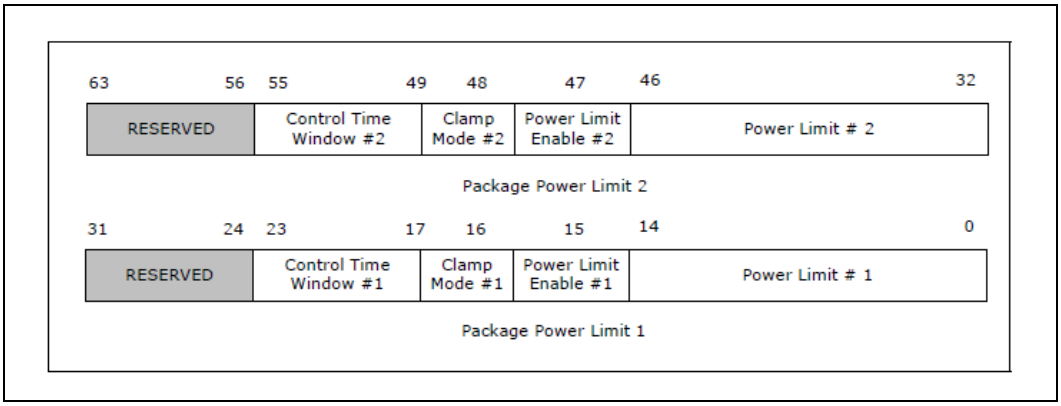



While Intel requires reading the accumulated energy data at least once every 16 seconds to ensure functional correctness, a more realistic polling rate recommendation is once every 100 ms for better accuracy. In general, as the power capability decreases, so will the minimum polling rate requirement. Intel recommends to tune the polling rate to reduce the potential impact on other power management features.

## 30.9.12 Package Power Limits For Multiple Turbo Modes (Index = 26 and 27)

This feature allows the PECI host to program two power limit values to support multiple turbo modes. The operating systems and drivers can balance the power budget using these two limits. Two separate PECI requests are available to program the lower and upper 32 bits of the power limit data shown in Figure 30-29, "Package Turbo Power Limit Data."

**Figure 30-29.Package Turbo Power Limit Data**




The units for the power limit and control time window are determined as per the package power SKU unit settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)," while the valid range for the power limit values are determined by the package power SKU settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)" and Section 30.9.3, "Package Power SKU Read (Index = 28 and 29)."

Setting the Clamp Mode bits is required to allow the cores to go into power states below what the operating system originally requested. The Power Limit Enable bits should be set to enable the power limiting function. Power limit values, Enable, and Clamp Mode bits can all be set in the same command cycle.

Intel recommends exclusive use of just one entity or interface, PECI for instance, to manage all processor package power limiting and budgeting needs. If PECI is being used to manage package power limiting activities, the BIOS should lock out all subsequent in-band package power limiting accesses by setting bit 31 of the MSR_PKG_POWER_LIMIT (MSR 610h) or TURBO_POWER_LIMIT (SoC sideband Port 04h, offset 7 and 8) register to 1.

Power Limit #1 is intended to limit processor power consumption to any reasonable value below TDP and defaults to TDP. Power Limit #1 values may be impacted by the processor heat sinks and system air flow. Processor Power Limit #2 can be used as appropriate to limit the current drawn by the processor to prevent any external power supply unit issues. Power Limit #2 should always be programmed to a value (typically 20%) higher than Power Limit #1 and has no default value associated with it.

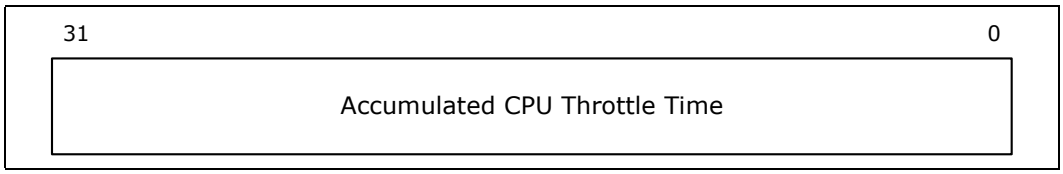
Though this feature is disabled by default and external programming is required to enable, initialize, and control the package power limit values and time windows, the processor package will still turbo to TDP if Power Limit #1 is not enabled or initialized. Control Time Window #1 values may be programmed to be within a range of 100 ms-10 seconds. Control Time Window #2 values should be in the range 40 ns to 10 ms.

The same conversion formula used for the DRAM Power Limiting feature (see Section DRAM Power Limit Data Read/Write) should be applied when programming the 'Control Time Window' in bits [23:17] and [55:49].

### 30.9.13 Package Power Limit Performance Status Read (Index = 8)

This service allows the PECI host to assess the performance impact of the currently active power limiting modes. The read return data contains the total amount of time for which the entire processor package has been operating in a power state that is lower than what the operating system originally requested. This information is tracked by a 32-bit counter that wraps around. See Figure 30-30, "Package Power Limit Performance Data." The unit for time is determined as per the package power SKU unit settings described in Section 30.9.2, "Package Power SKU Unit Read (Index = 30)."

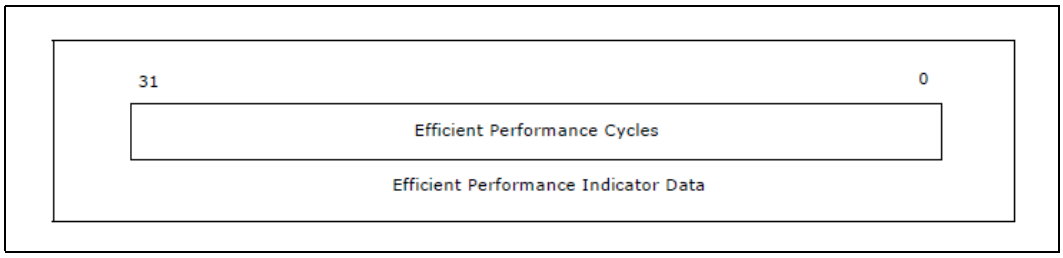
**Figure 30-30.Package Power Limit Performance Data**




### 30.9.14 Efficient Performance Indicator Read (Index = 6)

The Efficient Performance Indicator (EPI) Read provides an indication of the total number of productive cycles. Specifically, these are the cycles when the processor is engaged in any activity to retire instructions and as a result, consuming energy. Any power management entity monitoring this indicator should sample it at least once every 4 seconds to enable detection of wraparounds.

**Figure 30-31.Efficient Performance Indicator Data**



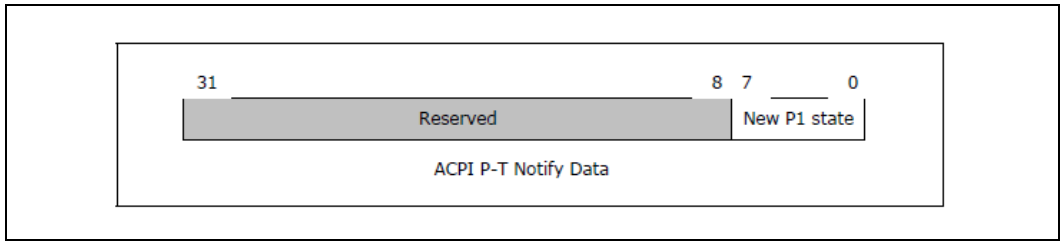
### 30.9.15 ACPI P-T Notify Write and Read (Index = 33)

This feature enables the processor turbo capability when used in conjunction with the PECI package RAPL or power limit. When the BMC sets the package power limit to a value below TDP, it also determines a new corresponding turbo frequency and notifies the OS using the 'ACPI Notify' mechanism as supported by the _PPC or performance present capabilities object. The BMC then notifies the processor PCU using the PECI 'ACPI P-T Notify' service by programming a new state that is one p-state below the turbo frequency sent to the OS via the _PPC method.

When the OS requests a p-state higher than what is specified in bits [7:0] of the PECI ACPI P-T Notify data field, the CPU will treat it as request for P0 or turbo. The PCU will use the IA32_ENERGY_PERFORMANCE_BIAS register settings to determine the exact extent of turbo. Any OS p-state request that is equal to or below what is specified in the PECI ACPI P-T Notify will be granted as long as the RAPL power limit does not impose a lower p-state. However, turbo will not be enabled in this instance even if there is headroom between the processor energy consumption and the RAPL power limit.

This feature does not affect the Thermal Monitor behavior of the processor nor is it impacted by the setting of the power limit clamp mode bit.

**Figure 30-32.ACPI P-T Notify Data**



## 30.9.16 Package PL3 (Index = 53)

PL3 is a package limit that is defined for the overall SoC in support of a PSU requirement of a strict duty cycle enforcement for points above PL2. This allows the PSU to be designed for application power draw, rather than virus.

**Figure 30-33.PL2 and PL3**

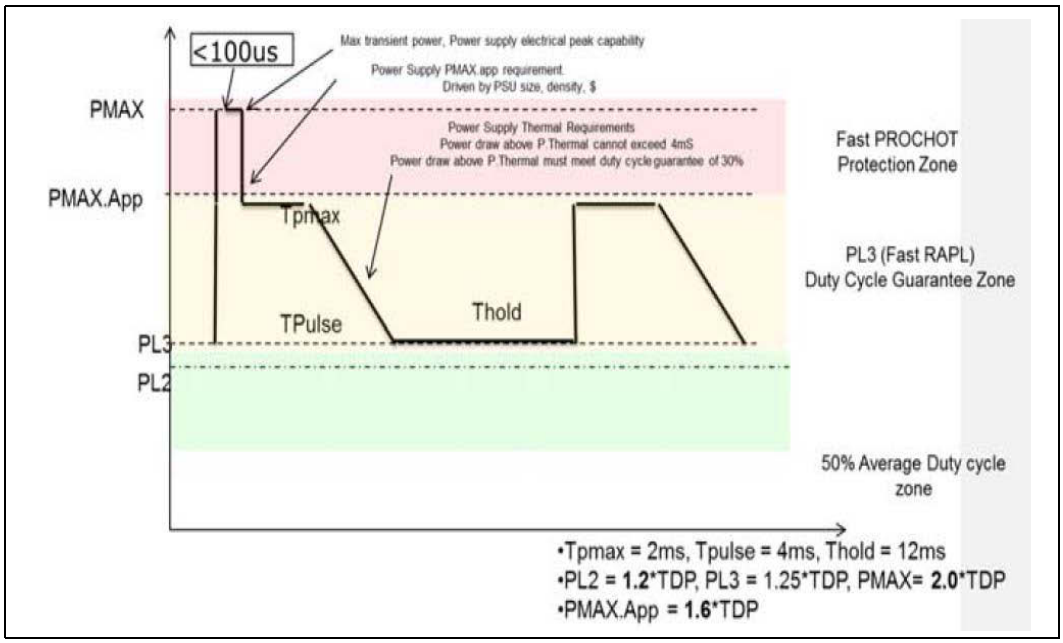
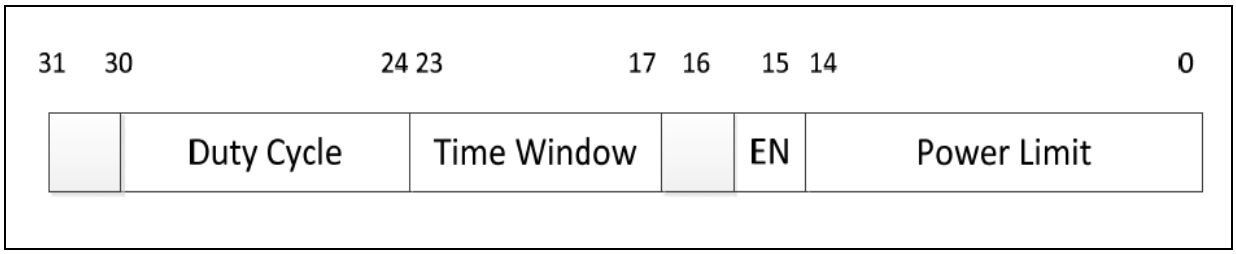



**Figure 30-34.PL3 Register Definition**



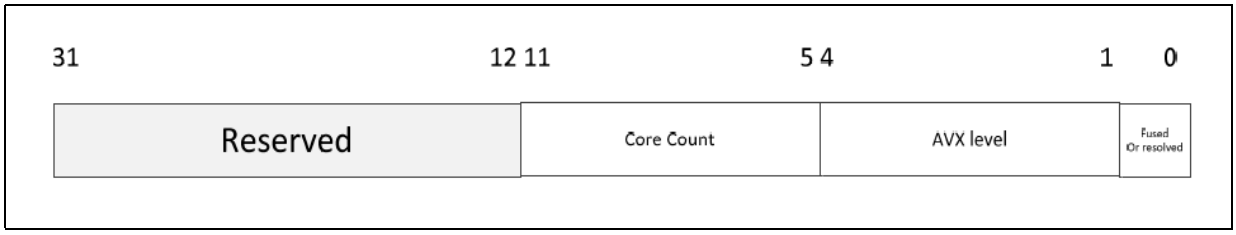

The defaults values are:

Power Limit: 1.25 TDP
Enable: Always set
Time Window: 8 ms
Duty-Cycle: 25%

## 30.9.17 Read Ratio Constraints (Index = 49)

Read constraints takes a single parameter that indicates the desired information

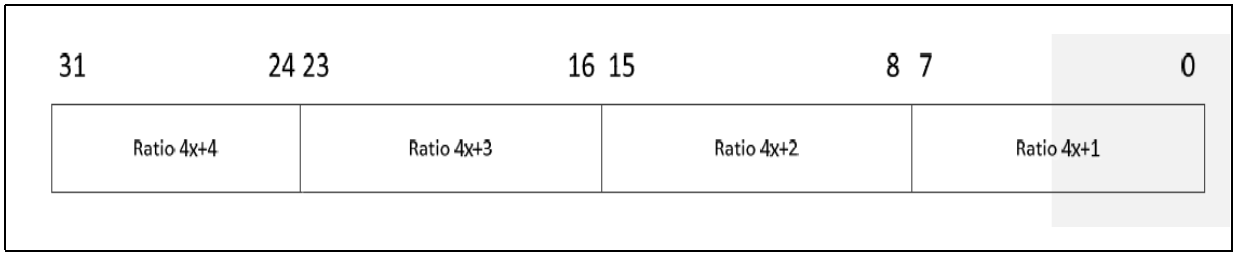
**Figure 30-35.Read Ratio Constraints Parameter**




Bit [0] indicates fused or resolved. 0:fused, 1: resolved.
Bit [4:1] indicate AVX level. 0h: non-AVX, 1h: AVX, 2h: AVX3.
Bits [11:5] indicate core count input x (0h - 7Fh).

The return value is shown in Figure 30-36, "Read Ratio Constraints Return Value."

**Figure 30-36.Read Ratio Constraints Return Value**

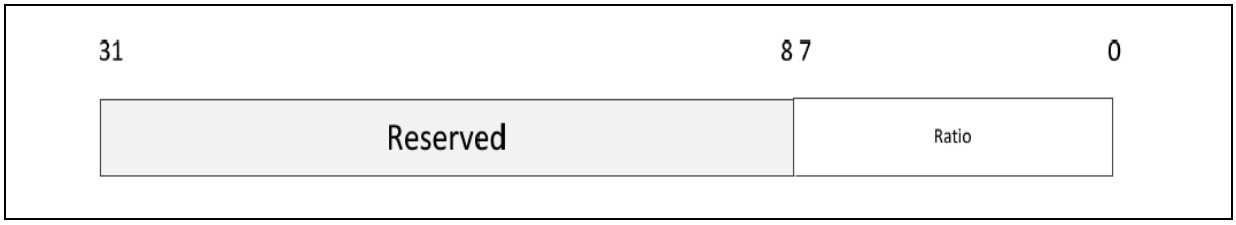



- If 4x + y, y = {1,2,3,4} is not a valid core count, returns ffh as the limit.

### 30.9.18 Read Ratio Limit

This read returns the ratio limit.

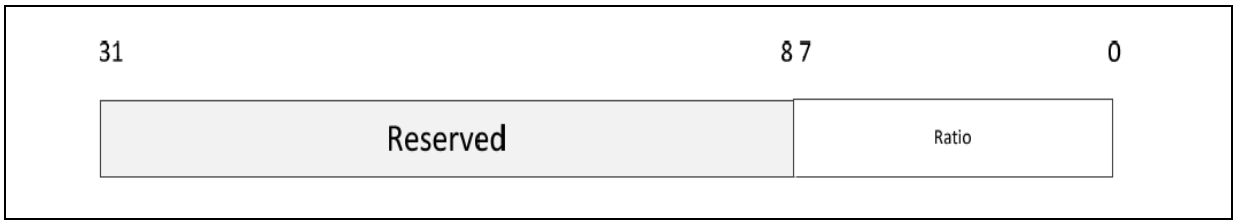
**Figure 30-37.Read Ratio Limit**




### 30.9.19 Write Ratio Limit

The ratio limit is defined in the data field.

**Figure 30-38.Write Ratio Limit**



## 30.10 Crash Dump

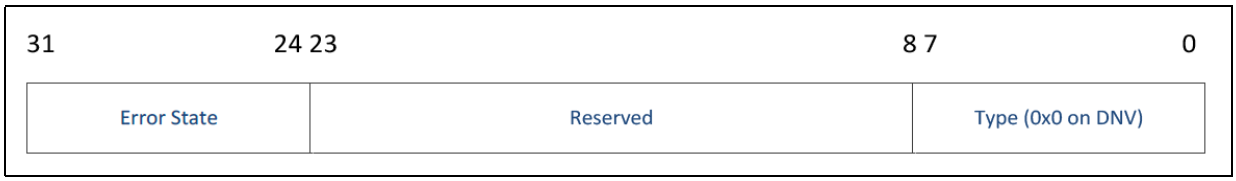
Crash Dump is implemented as a RdPkgConfig Command.

### 30.10.1 GetCrashDumpCapabilities (Index=51)

This feature returns the details of a crash dump. When parameter#0 is:

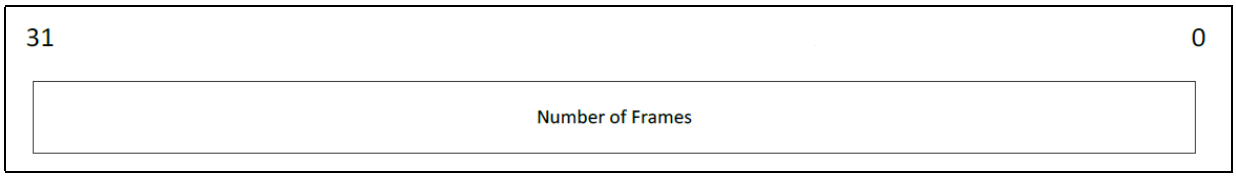
0x0 - returns the size info:

**Figure 30-39.Crash Dump Capabilities Details**




0x1 - returns the number of frames:

**Figure 30-40.Crash Dump Capabilities (Parameter0 = 0x1)**



### 30.10.2 GetCrashDumpFrame (Index = 52)

This feature gets a particular frame in the specified crash dump. When requesting the frame, the frame# is passed as a parameter. A maximum of 2^16 frames is only supported, starting at frame '0'.

**Figure 30-41.Request Parameters**

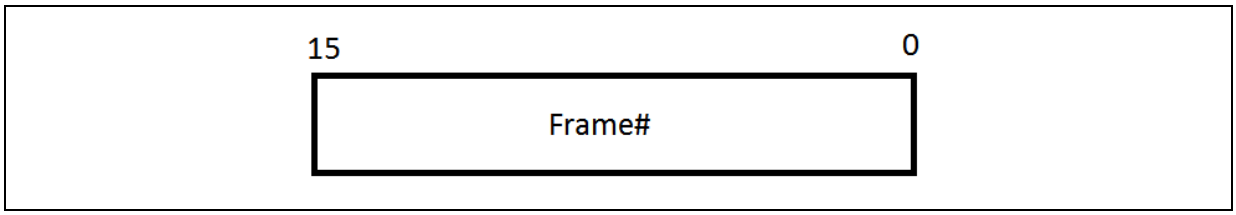
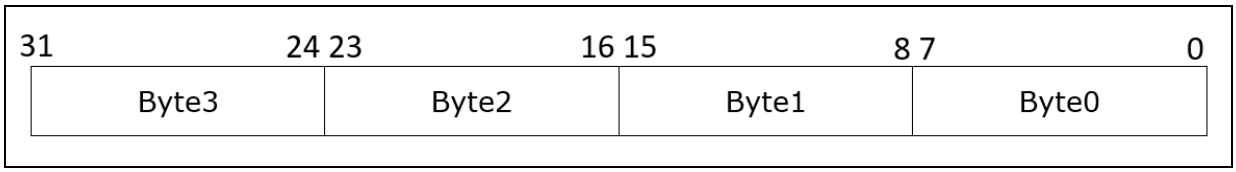



**Figure 30-42.Response Data for Requested Frame**



## 30.10.3 UnCore and PM Crash Dump Registers

The following are the registers visible in crash dump. The purpose of these registers is to enable customers to gather important system information when a crash happens in their system and to provide the information contained in these registers to Intel for analysis.

**Table 30-36. Response Data for Requested Items**

| Type | Source | Register | Size Each (B) | Granularity | Count | Total Size | Offset |
|---|---|---|---|---|---|---|---|
| HEADER | Block | ART | 8 | 1 | 1 | 8 | 0 |
| HEADER | PUNIT | Crashlog Version | 4 | 1 | 1 | 4 | 8 |
| HEADER | PUNIT | Reason for Record | 4 | 1 | 1 | 4 | 12 |
| HEADER | PUNIT | Punit Patch version | 4 | 1 | 1 | 4 | 16 |
| PADDING | | | | | | | |
| REG | BUNIT | RESERVED_1 | 4 | 1 | 1 | 4 | 32 |
| REG | BUNIT | RESERVED_2 | 4 | 1 | 1 | 4 | 36 |
| REG | AUNIT | RESERVED_3 | 4 | 1 | 1 | 4 | 40 |
| REG | DUNIT | RESERVED_4 | 4 | 1 | 1 | 4 | 44 |
| REG | DUNIT | RESERVED_5 | 4 | 1 | 1 | 4 | 48 |
| REG | DUNIT | RESERVED_6 | 4 | PER_DUNIT | 2 | 8 | 52 |
| REG | DUNIT | RESERVED_7 | 4 | PER_DUNIT | 2 | 8 | 60 |
| REG | DUNIT | RESERVED_8 | 4 | PER_DUNIT | 2 | 8 | 68 |
| REG | DUNIT | RESERVED_9 | 4 | PER_DUNIT | 2 | 8 | 76 |
| REG | DUNIT | RESERVED_10 | 4 | PER_DUNIT | 2 | 8 | 84 |
| REG | DUNIT | RESERVED_11 | 4 | PER_DUNIT | 2 | 8 | 92 |
| REG | DUNIT | RESERVED_12 | 4 | PER_DUNIT | 2 | 8 | 100 |
| REG | DUNIT | RESERVED_13 | 4 | PER_DUNIT | 2 | 8 | 108 |
| REG | DUNIT | RESERVED_14 | 4 | PER_DUNIT | 2 | 8 | 116 |
| REG | DUNIT | RESERVED_15 | 4 | PER_DUNIT | 2 | 8 | 124 |
| REG | SVID | RESERVED_16 | 4 | 1 | 1 | 4 | 132 |
| REG | SVID | RESERVED_17 | 4 | 1 | 1 | 4 | 136 |
| REG | SVID | RESERVED_18 | 4 | 1 | 1 | 4 | 140 |
| REG | SVID | RESERVED_19 | 4 | 1 | 1 | 4 | 144 |
| REG | TUNIT | STATUS/RESERVED_20 | 8 | PER_MODULE | 8 | 64 | 148 |
| REG | PUNIT | RESERVED_21 | 4 | 1 | 1 | 4 | 212 |
| REG | PUNIT | P_CR_MEM_THERM_STATUS | 4 | 1 | 1 | 4 | 216 |
| REG | PUNIT | P_CR_MEM_THERM_INT_STATUS | 4 | 1 | 1 | 4 | 220 |
| REG | PUNIT | P_CR_IA_PERF_LIMIT_REASONS | 4 | 1 | 1 | 4 | 224 |
| REG | BUNIT | RESERVED_22 | 4 | 1 | 1 | 4 | 228 |
| REG | BUNIT | RESERVED_23 | 4 | 1 | 1 | 4 | 232 |
| REG | PECI | RESERVED_24 | 4 | 1 | 1 | 4 | 236 |
| REG | PECI | RESERVED_25 | 4 | 1 | 1 | 4 | 240 |
| REG | PECI | RESERVED_26 | 4 | 1 | 1 | 4 | 244 |
| REG | USB | PORTLI5 | 4 | 1 | 1 | 4 | 248 |

### Table 30-36. Response Data for Requested Items

| Type | Source | Register | Size Each (B) | Granularity | Count | Total Size | Offset |
|---|---|---|---|---|---|---|---|
| REG | USB | PORTLI6 | 4 | 1 | 1 | 4 | 252 |
| REG | USB | PORTLI7 | 4 | 1 | 1 | 4 | 256 |
| REG | USB | PORTLI8 | 4 | 1 | 1 | 4 | 260 |
| REG | USB | USB2_PORT_STATE_REG | 8 | 2 | 2 | 16 | 264 |
| REG | USB | USB3_PORT_STATE_REG | 8 | 2 | 2 | 16 | 280 |
| REG | IOMMU | FSTS_REG_0_0_0_VTDBAR | 4 | 1 | 1 | 4 | 296 |
| REG | GBE | RESERVED_27 | 8 | PER_GBE | 2 | 16 | 300 |
| REG | GBE | RESERVED_28 | 8 | PER_GBE | 2 | 16 | 316 |
| REG | GBE | RESERVED_29 | 8 | PER_GBE | 2 | 16 | 332 |
| REG | GBE | RESERVED_30 | 8 | PER_GBE | 2 | 16 | 348 |
| REG | GBE | LINKS | 8 | PER_GBE | 2 | 16 | 364 |
| REG | GBE | RESERVED_31 | 8 | PER_GBE | 2 | 16 | 380 |
| REG | GBE | RESERVED_32 | 8 | PER_GBE | 2 | 16 | 396 |
| REG | GBE | RESERVED_33 | 8 | PER_GBE | 2 | 16 | 412 |
| REG | GBE | RESERVED_34 | 8 | PER_GBE | 2 | 16 | 428 |
| REG | GBE | RESERVED_35 | 8 | PER_GBE | 2 | 16 | 444 |
| REG | GBE | RESERVED_36 | 8 | PER_GBE | 2 | 16 | 460 |
| REG | GBE | RESERVED_37 | 8 | PER_GBE | 2 | 16 | 476 |
| PADDING | | | | | | 292 | |
| COREPMA | GLM | RESERVED_38 | 4 | PER_MODULE | 8 | 32 | 768 |
| COREPMA | GLM | RESERVED_39 | 4 | PER_MODULE | 8 | 32 | 800 |
| COREPMA | GLM | RESERVED_40 | 4 | PER_MODULE | 8 | 32 | 832 |
| COREPMA | GLM | RESERVED_41 | 4 | PER_MODULE | 8 | 32 | 864 |
| COREPMA | GLM | RESERVED_42 | 4 | PER_MODULE | 8 | 32 | 896 |
| COREPMA | GLM | RESERVED_43 | 4 | PER_MODULE | 8 | 32 | 928 |
| COREPMA | GLM | RESERVED_44 | 4 | PER_MODULE | 8 | 32 | 960 |
| COREPMA | GLM | RESERVED_45 | 4 | PER_MODULE | 8 | 32 | 992 |
| PADDING | | | | | | 32 | |
| IOREG | PUNIT | RESERVED_46 | 4 | N/A | 0 | 0 | 1024 |
| IOREG | PUNIT | RESERVED_47 | 4 | PER_CORE | 16 | 64 | 1024 |
| IOREG | PUNIT | RESERVED_48 | 4 | PER_MODULE | 8 | 32 | 1088 |
| IOREG | PUNIT | RESERVED_49 | 4 | N/A | 0 | 0 | 1120 |
| IOREG | PUNIT | RESERVED_50 | 4 | N/A | 0 | 0 | 1120 |
| IOREG | PUNIT | RESERVED_51 | 4 | 1 | 1 | 4 | 1120 |
| IOREG | PUNIT | RESERVED_52 | 4 | N/A | 0 | 0 | 1124 |
| IOREG | PUNIT | RESERVED_53 | 4 | PER_CORE | 16 | 64 | 1124 |
| IOREG | PUNIT | RESERVED_54 | 4 | PER_CORE | 16 | 64 | 1188 |
| IOREG | PUNIT | RESERVED_55 | 4 | N/A | 0 | 0 | 1252 |
| IOREG | PUNIT | RESERVED_56 | 4 | N/A | 0 | 0 | 1252 |

### Table 30-36. Response Data for Requested Items

| Type | Source | Register | Size Each (B) | Granularity | Count | Total Size | Offset |
|---|---|---|---|---|---|---|---|
| IOREG | PUNIT | RESERVED_57 | 4 | 1 | 1 | 4 | 1252 |
| IOREG | PUNIT | RESERVED_58 | 4 | N/A | 0 | 0 | 1256 |
| IOREG | PUNIT | RESERVED_59 | 4 | N/A | 0 | 0 | 1256 |

## 30.10.4 Core Crash Dump

### 30.10.4.1 Overview

Core crash dump provides the SoC with the ability to dump the core registers (including the SQ) when a 3-strike crash takes place. To provide this capability, RdPkgConfig has been implemented to access the core crash dump.

### 30.10.4.2 PECI Command Structure

RdPkgCOnfig is used with index 0x51. The parameter field is divided into 2 bytes. Byte0 is always 0x2 and Byte1 is the coreindex.

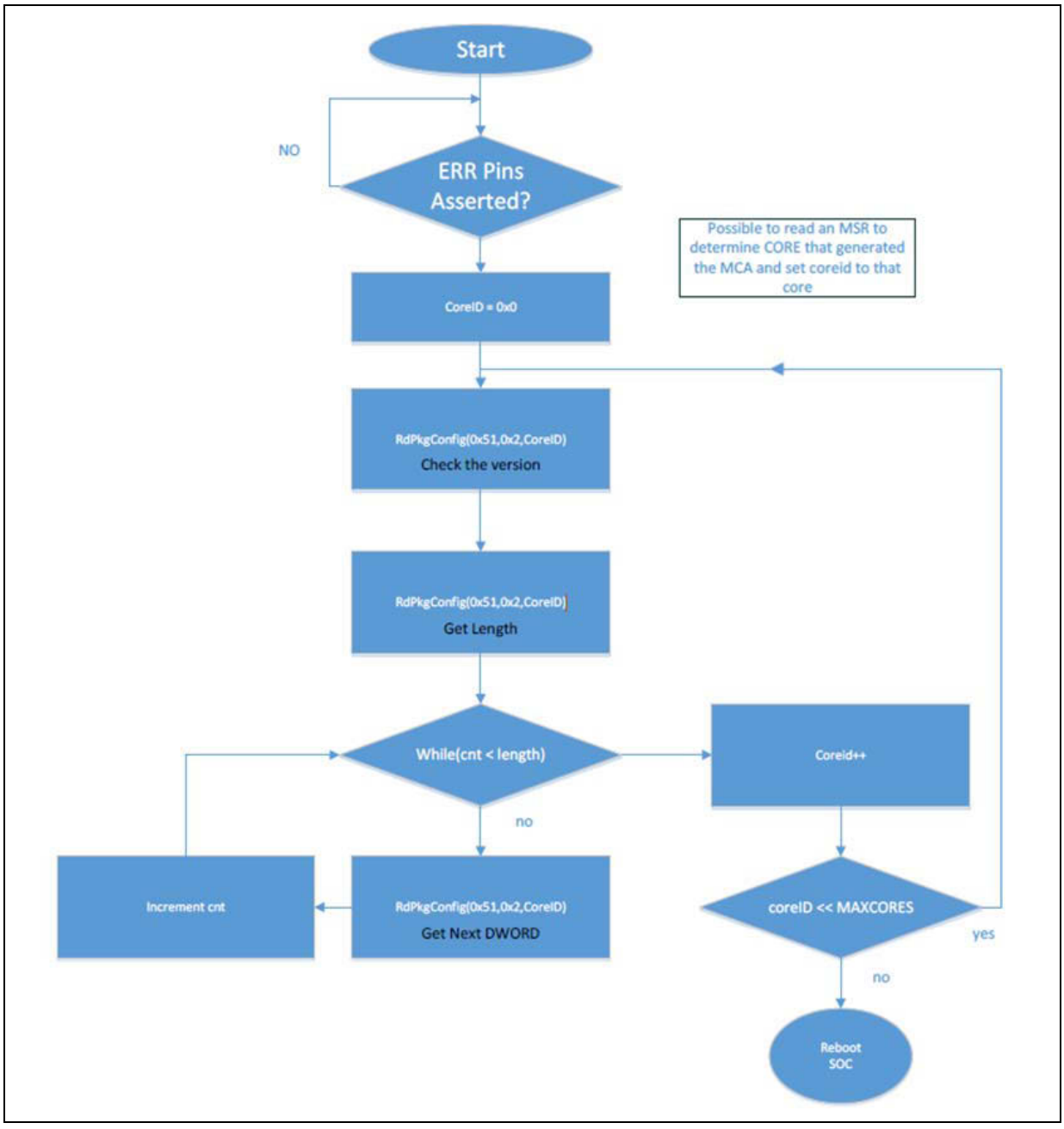
**Table 30-37. RdPkgConfig() PECI Proxy Block Write**

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Write) | Write Address | SMBus Command | 0x4C | BMC | |
| Command Code | PECI Mode command code | | 0x62 | | |
| Byte Count | N | Control Data | 0x09 | | |
| Byte 1 | SMBus-PECI Handshake Control | | 0x00 | | Value = 0x00<br>No AW FCS Required |
| Byte 2 | PECI Client Address | PECI Command | 0x30 | | Socket ID. Always 0x30 for the SoC. |
| Byte 3 | PECI Write Length | | 0x05 | | |
| Byte 4 | PECI Read Length | | {0x02, 0x3, 0x5} | | |
| Byte 5 | RdPkgConfig() Command Code | | 0xA1 | | |
| Byte 6 | Host ID and Retry | | 0x00 | | |
| Byte 7 | Index | | 51 | | See Section 30.7.11, "MCA Access via RdEndPointConfig()" and Section 30.9, "CPU Thermal and Power Optimization Capabilities" for the supported capabilities and their returned data structures. |
| Byte 8 | Parameter LSB | | 0x2 | | |
| Byte 9 | Parameter MSB | | coreID 0-15 | | |

## Table 30-38. RdPkgConfig() PECI Proxy Block Read

| SMBus | Function | | Value | Data Source | Comment |
|---|---|---|---|---|---|
| Slave Address (Command Phase) | Write Address | SMBus Command | 0x4B | BMC | |
| SMBus Read Command Code | SMBus Read Code | | 0x40 | | |
| Slave Address (Data Phase) | Read Address | | 0x4B | | |
| Byte Count | N | Control Data | {0x01, 0x02, 0x04, 0x05, 0x07} | SoC | N = 0x01 for busy state<br>• Byte 1 = 0x01<br>N = 0x02 for transaction errors<br>• Byte 1 = 0x02<br>• Byte 2 = Error Code defined in Table 30-6, "PECI Proxy Read."<br>N = {0x4,0x5,0x7} indicates a successful PECI transaction<br>• Byte 1 = 0x00<br>• Byte 2 = 0x00<br>• Byte [7:3] = PECI response data<br>N = {0x04, 0x05, 0x07} value is determined by the number of PECI Response Data bytes returned in Byte [7:4] on the SMBus.<br>N Value:<br>• 0x04 = 1 data byte<br>• 0x05 = 2 data bytes<br>• 0x07 = 4 data bytes |
| Byte 1 | Status Byte | | CMD_STAT | | See Byte [1] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 2 | Error Byte | | ERR_CODE | | See Byte [2] of defined in Table 30-6, "PECI Proxy Read." |
| Byte 3 | PECI Transaction Status | PECI Response Data | PECI Completion Code | | Completion Code decode:<br>• 0x40 = Command successful<br>• 0x80 = Response time-out<br>• 0x90 = Illegal command |
| Byte 4 | Data 1 (LSB) | | Data = 1, 2, or 4 bytes | | See Section 30.7.11, "MCA Access via RdEndPointConfig()" and Section 30.9, "CPU Thermal and Power Optimization Capabilities" for the supported capabilities and their returned data structures. |
| Byte 5 | Data 2 | | | | |
| Byte 6 | Data 3 | | | | |
| Byte 7 | Data 4 (MSB) | | | | |

### 30.10.4.3 Flow


Start
NO
ERR Pins Asserted?
Possible to read an MSR to determine CORE that generated the MCA and set coreid to that core
CoreID = 0x0
RdPkgConfig(0x51,0x2,CoreID)
Check the version
RdPkgConfig(0x51,0x2,CoreID)
Get Length
While(cnt < length)
Coreid++
no
Increment cnt
RdPkgConfig(0x51,0x2,CoreID)
Get Next DWORD
coreID << MAXCORES
yes
no
Reboot SOC

### 30.10.4.4 Core Crash Details

**Table 30-39. Core Crash Details**

| Field | Category | Size (bytes) |
|---|---|---|
| **Module0 Regs** | | |
| cpu.module0.crashlog_version | Version - 0x10010 | 4 |
| cpu.module0.payload_size | Bits 31:16 = Core Size;<br>Bits 15:0 = Module Size | 4 |
| cpu.scp_cr_who_am_i_info | Only includes bits 14:0 | 4 |
| cpu.module0.patch_revision | Misc Info | 4 |
| cpu.module0.l2_cr_mci_status | Module MCA Regs | 8 |
| cpu.module0.bus_cr_mci_status | Module MCA Regs | 8 |
| cpu.module0.bus_cr_mci_addr | Module MCA Regs | 8 |
| **Module0 XQ** | | |
| cpu.module0.XQ_0 | Header | 8 |
| cpu.module0.XQ_1 | Header | 8 |
| cpu.module0.XQ_2 | Header | 8 |
| cpu.module0.XQ_3 | Header | 8 |
| cpu.module0.XQ_4 | Header | 8 |
| cpu.module0.XQ_5 | Header | 8 |
| cpu.module0.XQ_6 | Header | 8 |
| cpu.module0.XQ_7 | Header | 8 |
| cpu.module0.XQ_8 | Header | 8 |
| cpu.module0.XQ_9 | Header | 8 |
| cpu.module0.XQ_10 | Header | 8 |
| cpu.module0.XQ_11 | Header | 8 |
| cpu.module0.XQ_12 | Header | 8 |
| cpu.module0.XQ_13 | Header | 8 |
| cpu.module0.XQ_14 | Header | 8 |
| cpu.module0.XQ_15 | Header | 8 |
| **Module 0 Intel Debug Information** | | |
| cpu.module0.intel_debug0 | Module Debug Info | 8 |
| cpu.module0.intel_debug1 | Module Debug Info | 8 |
| **Core0 Header** | | |
| cpu.module0.core0.timestamp | Header | 8 |
| cpu.module0.core0.crashlog_reason | Header | 4 |
| **Core0 Arch State** | | |
| cpu.module0.core0.thread0.x86_cr_cr0 | Arch State | 4 |
| cpu.module0.core0.thread0.scp_cr_x86_cr2 | Arch State | 8 |
| cpu.module0.core0.thread0.x86_cr_cr3 | Arch State | 8 |
| cpu.module0.core0.thread0.x86_cr_cr4 | Arch State | 4 |

### Table 30-39. Core Crash Details

| | | |
|---|---|---|
| cpu.module0.core0.thread0.ia32_cr_efer | Arch State | 4 |
| cpu.module0.core0.lip | Arch State | 8 |
| cpu.module0.core0.eflags | Arch State | 4 |
| cpu.module0.core0.cs_selector | Arch State | 2 |
| cpu.module0.core0.ss_selector | Arch State | 2 |
| cpu.module0.core0.ds_selector | Arch State | 2 |
| cpu.module0.core0.es_selector | Arch State | 2 |
| cpu.module0.core0.fs_selector | Arch State | 2 |
| cpu.module0.core0.gs_selector | Arch State | 2 |
| cpu.module0.core0.rax | GP Regs | 8 |
| cpu.module0.core0.rcx | GP Regs | 8 |
| cpu.module0.core0.rdx | GP Regs | 8 |
| cpu.module0.core0.rbx | GP Regs | 8 |
| cpu.module0.core0.rsp | GP Regs | 8 |
| cpu.module0.core0.rbp | GP Regs | 8 |
| cpu.module0.core0.rsi | GP Regs | 8 |
| cpu.module0.core0.rdi | GP Regs | 8 |
| cpu.module0.core0.r8 | GP Regs | 8 |
| cpu.module0.core0.r9 | GP Regs | 8 |
| cpu.module0.core0.r10 | GP Regs | 8 |
| cpu.module0.core0.r11 | GP Regs | 8 |
| cpu.module0.core0.r12 | GP Regs | 8 |
| cpu.module0.core0.r13 | GP Regs | 8 |
| cpu.module0.core0.r14 | GP Regs | 8 |
| cpu.module0.core0.r15 | GP Regs | 8 |
| **Core0 MC Regs** | | |
| cpu.module0.core0.ic_cr_mci_status | Core MCA Regs | 8 |
| cpu.module0.core0.ic_cr_mci_addr | Core MCA Regs | 8 |
| cpu.module0.core0.mec_cr_mci_status | Core MCA Regs | 8 |
| cpu.module0.core0.mec_cr_mci_addr | Core MCA Regs | 8 |
| **Core0 Intel Debug Information** | | |
| cpu.module0.core0.intel_debug0 | Debug Info | 8 |
| cpu.module0.core0.intel_debug1 | Debug Info | 8 |

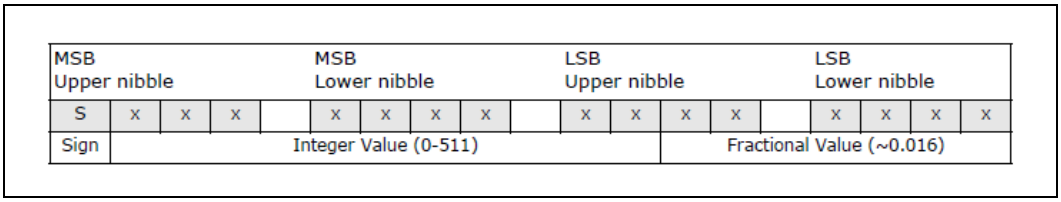
# 30.11 DTS Temperature Data

## 30.11.1 PECI Device Temp Data

When accessed using the PECI GetTemp() command, the temperature is formatted in a 16-bit, 2s complement value representing a number of 1/64 °C.

See Figure 30-43, "PECI Device Temp [15:0] Format - Temperature Sensor Data." This format allows temperatures in a range of +/-512 °C to be reported to approximately a 0.016 °C resolution.

**Figure 30-43.PECI Device Temp [15:0] Format - Temperature Sensor Data**




## 30.11.2 Interpretation

The resolution of the processor Digital Thermal Sensor (DTS) is approximately 1 °C, which can be confirmed by a performing a Read Model-Specific Register (RDMSR) Intel® architecture instruction to the IA32_THERM_STATUS MSR (19Ch) where it is architecturally defined.

*Note:* The MSR read will return only bits [13:6] of the PECI temperature sensor data defined in Figure 30-43, "PECI Device Temp [15:0] Format - Temperature Sensor Data." PECI temperatures are sent through a configurable low-pass filter prior to delivery in the GetTemp() response data. The output of this filter produces temperatures at the full 1/64 °C resolution even though the DTS itself is not this accurate.

Temperature readings from the processor are always negative in a 2s complement format, and imply an offset from the processor $T_{J\text{-}MAX}$ (PECI = 0). For example, if the processor $T_{J\text{-}MAX}$ is 100 °C, a PECI thermal reading of -10 implies that the processor is running at approximately 10 °C below $T_{J\text{-}MAX}$ which would be 90 °C. PECI temperature readings are not reliable at temperatures above $T_{J\text{-}MAX}$ since the processor is outside its operating range; therefore, the PECI temperature readings are never positive.

The changes in PECI data counts are approximately linear in relation to changes in temperature in degrees Celsius. A change of 1 in the PECI count represents roughly a temperature change of 1 °C. This linearity is approximate and cannot be guaranteed over the entire range of PECI temperatures, especially as the offset from the maximum PECI temperature (zero) increases.

### 30.11.3 Temperature Filtering

The processor Digital Thermal Sensor (DTS) provides an improved capability to monitor device hot spots, which inherently leads to more varying temperature readings over short time intervals. Coupled with the fact that typical fan speed controllers may only read temperatures at 4 Hz, the thermal readings need to reflect the thermal trends and not instantaneous readings. Therefore, PECI supports a configurable low-pass temperature filtering function that is expressed by the equation:

$$T_N = (1-\alpha) * T_{N-1} + \alpha * T_{SAMPLE}$$

where $T_N$ and $T_{N-1}$ are the current and previous averaged PECI temperature values respectively, $T_{SAMPLE}$ is the current PECI temperature sample value and the variable "α" = $1/2^X$, where *X* is the Thermal Averaging Constant that is programmable as described in Section 30.9.8, "Package Thermal Status Read / Clear (Index = 20)."

### 30.11.4 Reserved Values

Several values well out of the operational range are reserved to signal temperature sensor errors. These are shown in Table 30-40, "Error Codes."

**Table 30-40. Error Codes**

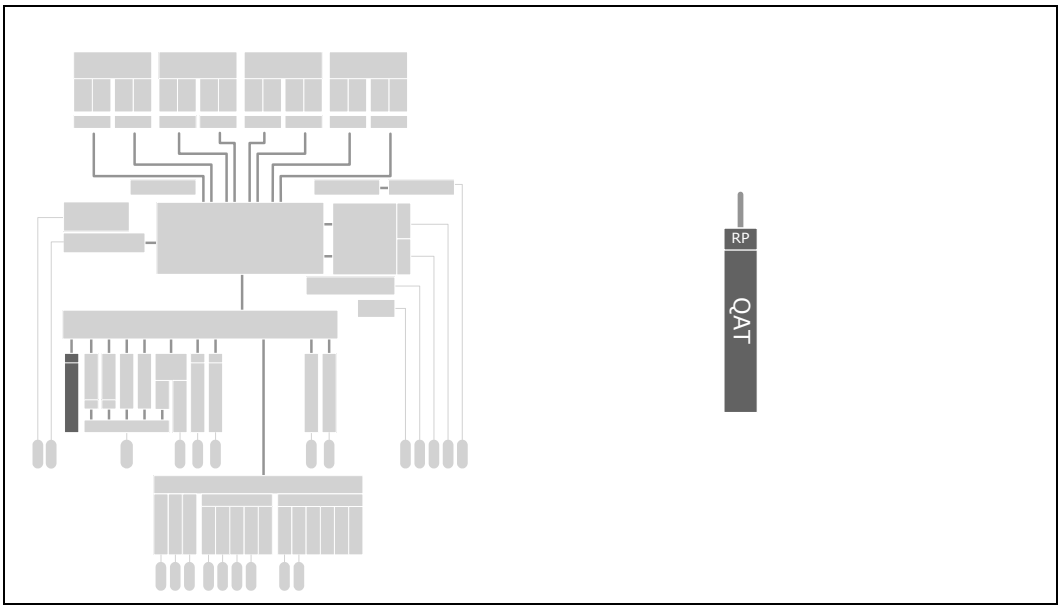
| Error Codes | Description |
|---|---|
| 0x8000 | General sensor error |
| 0x8001 | Reserved |
| 0x8002 | Sensor is operational, but has detected a temperature below its operational range. |
| 0x8003 | Sensor is operational, but has detected a temperature above its operational range. |
| 0x8004 – 0x81FF | Reserved |

§ §

# 31 Intel® QuickAssist Technology (Intel® QAT)

The integrated Intel® QuickAssist Technology (Intel® QAT) provides acceleration functions that can be used by the IA cores. The acceleration services can be accessed by a standard PCIe* driver. To enhance communication between the IA and the Intel® QAT, the Intel® QAT implements rings via the Ring controller to route messages between them.

**Figure 31-1. What is Covered in This Chapter**




Applications running on the IA core(s) can make use of the acceleration by calling the Intel® QuickAssist Technology Application Programming Interfaces (APIs). The APIs communicate with the Intel® QuickAssist Technology hardware via PCI configuration space access and assisted rings stored in system memory.

Refer to the *Intel® QuickAssist Technology API Programmer's Guide, incorporated herein by reference, and available at:*

- https://01.org/packet-processing/intel%C2%AE-quickassist-technology-drivers-and-patches.

## 31.1 Signal Descriptions

There are no signal pins or signal names associated with this chapter.

## 31.2 Features

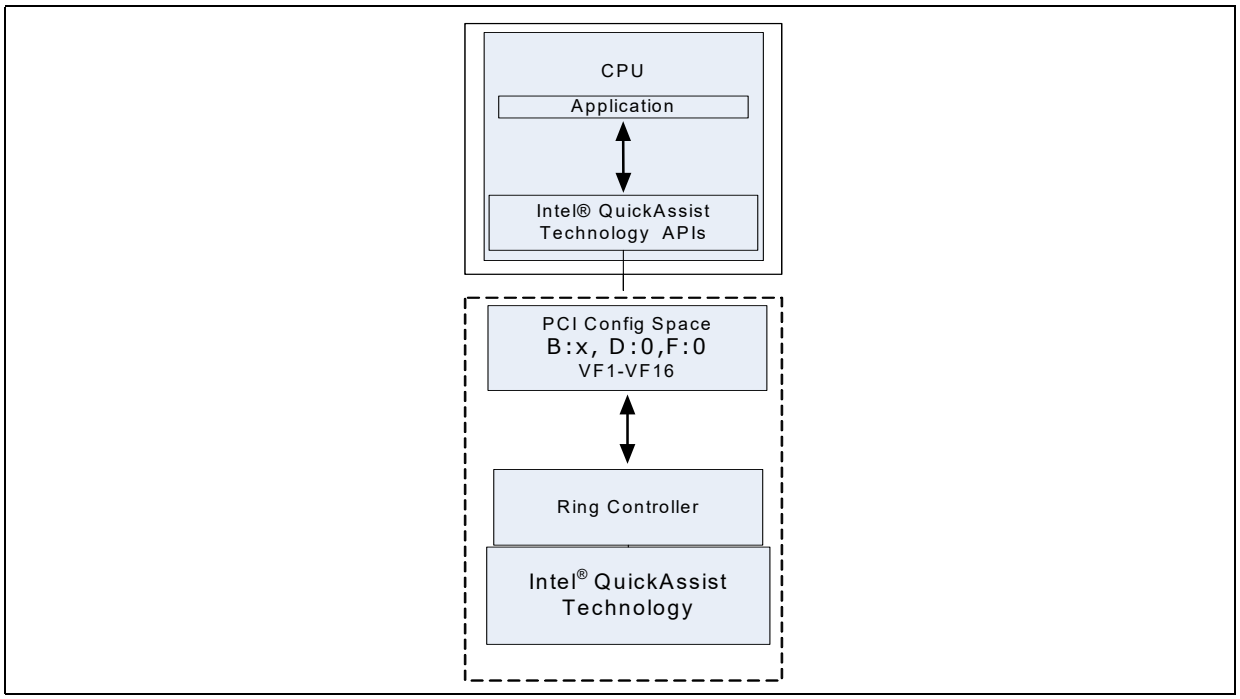
This product contains Intel® QuickAssist Technology hardware version 1.7.

- Symmetric cryptographic functions
  - Cipher operation
  - Hash/authenticate operation
  - Cipher-hash combined operation
  - Key derivation operation
- Public key functions
  - RSA operation
  - Diffie-Helman operation
  - Digital signature standard operation
  - Key derivation operation
  - Elliptic curve cryptography: ECDSA and ECDH
- Compression/Decompression
  - Deflate

## 31.3 Usage Model

Intel® QAT is used by applications running on the IA cores to accelerate and offload processing. Intel® QAT appears to software as a PCIe endpoint and is used as a look-aside co-processor.

**Figure 31-2. Intel® QAT Usage Block Diagram**




The following steps list a high-level overview of the usage.

1. The application requires Intel® QAT service (e.g., decrypting a packet using AES). It requests service by calling an API with appropriate parameters.
2. The Intel® QAT device driver formats a request message and places it onto ring-in-host memory, then notifies the Intel® QAT.
3. The Intel® QAT reads the request message from the Ring controller.
4. The Intel® QAT carries out the request.
5. When the process is done, write response message to the ring-in-host memory.
6. Interrupt the CPU via a Message Signaled Interrupt (MSI), or a user application can poll for completions.

*Note:* Refer to the *Intel® QuickAssist Technology API Programmer's Guide* for information on the Intel® QuickAssist Technology APIs available at:

- https://01.org/packet-processing/intel%C2%AE-quickassist-technology-drivers-and-patches.

### 31.3.1 Ring Controller

The Ring controller provides communication between the IA cores and the Intel® QuickAssist Integrated Accelerator.

- Implements 256 rings for communications between the CPU and the Intel® QAT.
- Organized in 16 bundles of 16 rings each.
- Each bundle supports one interrupt to the endpoint (EP).

### 31.3.2 Single-Root I/O Virtualization (SR-IOV)

A virtualized platform enables system hardware resources to be shared among multiple guest operating systems (OS). Each guest OS on the virtual platform operates under the assumption that it has complete control of all platform hardware resources. An I/O device that supports I/O virtualization provides the ability for it to be shared simultaneously among multiple guest operating systems.

The Intel® QAT EP implements support for single-root I/O virtualization for function 0 and enables a usage model that allows up to 16 guest operating systems simultaneous access to its resources.

SR-IOV is a PCI-SIG* I/O Virtualization (IOV) specification which in conjunction with system virtualization technologies like VT-d and VT-x, allows multiple guest operating systems running simultaneously within a single platform to natively share PCIe devices.

The SR-IOV capability structure is discovered and configured using the well-defined standard PCI enumeration process.

- Implements an SR-IOV extended capability record.
- One Physical Function (PF): Supports INTx, MSI, and MSI-X.
- 16 Virtual Functions (VF): Supports ARI. Implements MSI only. Does not support INTx messages or MSI-X.
- EP does not support the following optional IOV features:
  - Multi-root IOV
  - VF migration
- Ring access:
  - PF can access all 16 ring bundles.
  - Each VF can access one bundle.

*Note:* ARI must be enabled to use any Virtual Functions (VF).

#### 31.3.2.1 I/O Virtualization Models

An I/O device such as the Intel® QAT can be shared amongst multiple guest operating systems in a virtual platform using several different hardware, software, and a combination of software and hardware techniques.

#### 31.3.2.2 Sharing via Virtual Machine Monitor (VMM)

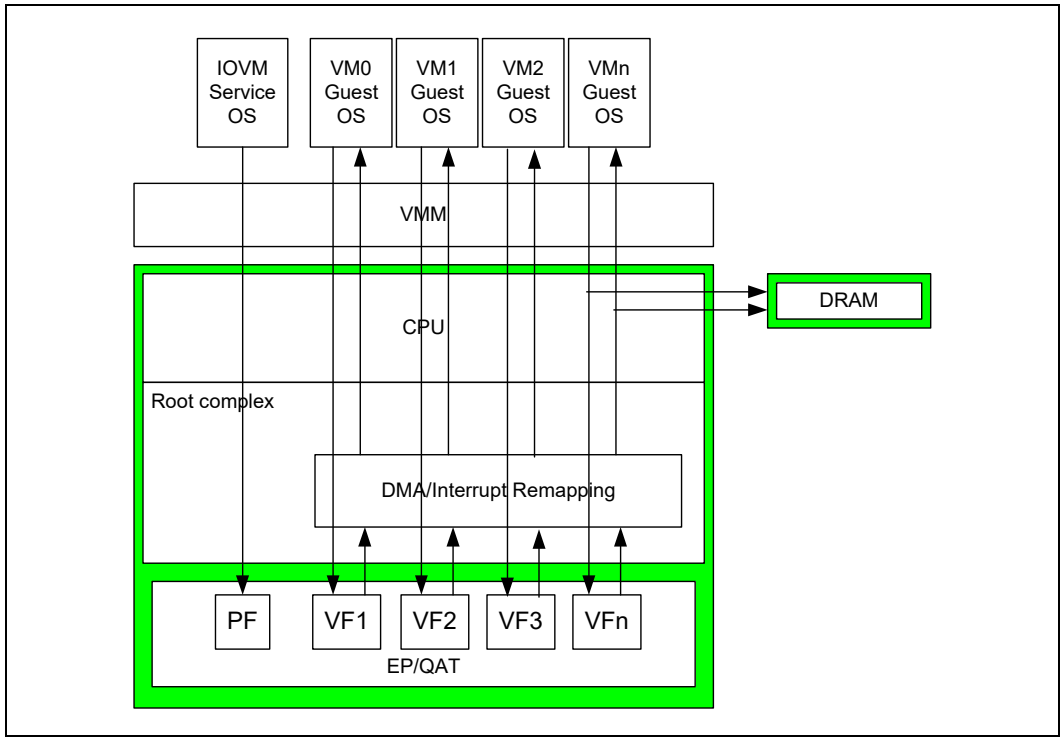
The I/O devices in this mode are assigned to the VMM/hypervisor. The VMM will expose a synthetic device to the Virtual Machines (VMs) and could provide a paravirtualized driver that will run in VM. All accesses from the VM using the paravirtualized driver are intercepted by the VMM. The VMM will then use the device driver to access the I/O device on behalf of the VM. In this form of I/O sharing, there is little or no hardware support in the root complex or I/O device for virtualization.

### 31.3.2.3 Direct Assignment of EP with SR-IOV

The VMM can take advantage of the Intel® QAT EP with SR-IOV support in the hardware and directly assign up to 16 VMs to the 16 VFs as shown in Figure 31-3, "Direct I/O Assignment Model with IOV Support." Each VM can independently access the services of the Intel® QAT.

Configuration and initialization of the Intel® QAT will be through the I/O Virtual Machine (IOVM). All error messages and interrupts not related to the ring status will be directed towards the IOVM (i.e., serviced by the IOVM).

**Figure 31-3. Direct I/O Assignment Model with IOV Support**



### 31.3.3 Ring Bundle Mapping

Each VF will be statically allocated one bundle (16 rings). A guest OS that is directly assigned to a VF will have access to only the rings associated with the bundle the VF is assigned to as shown in Table 31-1, "Ring Assignment to Intel® QAT Virtual Functions."

All errors and interrupts generated by the rings will use the Ring ID (RID) associated with that bundle.

**Table 31-1. Ring Assignment to Intel® QAT Virtual Functions**

| Ring | Virtual Function Assignment | Function Number | Ring CSR PCI Offset Begin |
|---|---|---|---|
| Bundle 0 | VF1 | 8 | 00000h |
| Bundle 1 | VF2 | 9 | 01000h |
| Bundle 2 | VF3 | 10 | 02000h |
| Bundle 3 | VF4 | 11 | 03000h |
| Bundle 4 | VF5 | 12 | 04000h |
| Bundle 5 | VF6 | 13 | 05000h |
| Bundle 6 | VF7 | 14 | 06000h |
| Bundle 7 | VF8 | 15 | 07000h |
| Bundle 8 | VF9 | 16 | 08000h |
| Bundle 9 | VF10 | 17 | 09000h |
| Bundle 10 | VF11 | 18 | 0A000h |
| Bundle 11 | VF12 | 19 | 0B000h |
| Bundle 12 | VF13 | 20 | 0C000h |
| Bundle 13 | VF14 | 21 | 0D000h |
| Bundle 14 | VF15 | 22 | 0E000h |
| Bundle 15 | VF16 | 23 | 0F000h |

***Notes:***
1. The PF can view and access all bundles.
2. Interrupts and memory accesses generated by each of the bundles via ring commands will use the associated bundle's RID.

#### 31.3.3.1 VF Interrupts

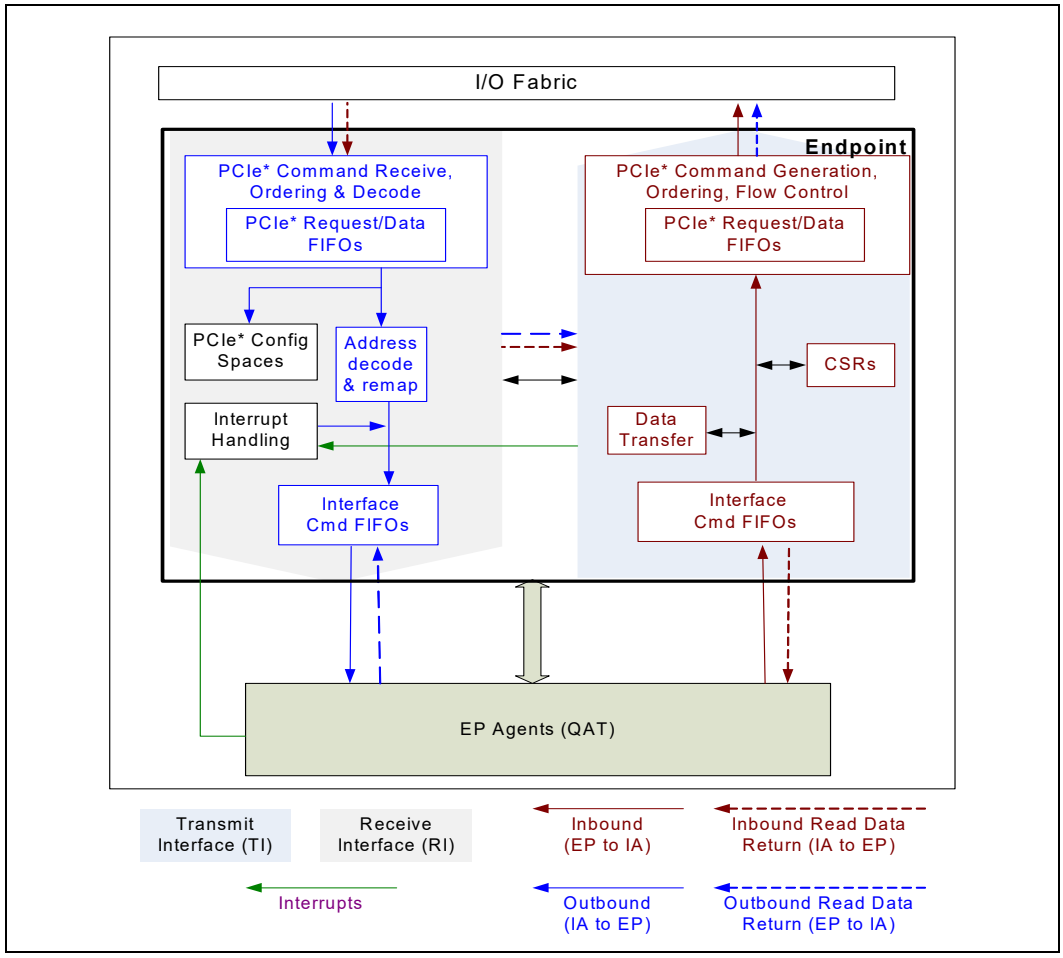
Each VF will support only MSI (one vector). A VF supports functional interrupts generated by the Ring controller. INTx is not applicable for VF per the PCI Express* SR-IOV specification.

## 31.3.4 PCIe Endpoint (EP) Function

The Intel® QAT will show up as a PCIe device to the software running on the IA cores. The Intel® QAT is an Endpoint Device/Function that is integrated in the SoC and connected to a dedicated Root Port.

Figure 31-4, "EP Functional Description Block Diagram" shows the functional description block diagram of the PCIe EP interface.

**Figure 31-4. EP Functional Description Block Diagram**




### 31.3.4.1 Transmit Interface (TI)

TI is responsible for converting EP agent commands toward system memory and Outbound (OB) completions into the PCIe transactions. The TI connects to the inbound port of the PCIe root complex.

### 31.3.4.2 Receive Interface (RI)

The Receive Interface (RI) is responsible for converting OB PCIe commands from the IA cores into the EP agents' commands. Downstream read completions for PCIe commands initiated by the TI are also received by the RI.

### 31.3.4.3 PCIe EP Functions

RI connects to the OB port of the PCIe root complex. Table 31-2, "PCIe Commands Supported by RI" lists the RI supported PCIe transactions. The EP only supports requests that are aligned on Dword address boundaries.

**Table 31-2. PCIe Commands Supported by RI**

| PCIe* Command | Description | Source | EP Attributes | Comments |
|---|---|---|---|---|
| MRd32/MRd64 | Memory Read | IA reading EP MMIO region, PCIe* extended configuration space | 4B | |
| MWr32/MWr64 | Memory Write | IA writing to EP MMIO region, PCIe extended configuration space | 4-8B | Note 2 |
| CfgRd0 | Type 0 Configuration Read | IA reading EP Configuration register | 1-4B | |
| CfgWr0 | Type 0 Configuration Write | Non-Posted<br>IA writing to EP Configuration register | 1-4B | Note 2 |
| CmplD | Completion with Data | Read completion with data for a read command initiated by TI | 4B-256B | |

***Notes:***
1. Although the EP is capable of handling 8B requests, Intel® QAT does not support any target that supports 8B requests. Therefore, software should not attempt any 8B requests.
2. This is a non-posted transaction. RI will generate a completion without data after the configuration write is completed. The completion will be routed via a RI-TI private bus to the TI. The TI will initiate the PCIe completion packet.
3. The RI will generate an Unsupported Request (UR) completion if it receives any PCIe commands from IA that are not listed in this table.

### 31.3.4.4 EP Function Mapping

The EP RI implements a single PCIe device with SR-IOV, and Table 31-3, "PCIe EP Function Mapping" shows the mapping of the EP functionality. When SR-IOV is enabled, the RI exposes additional virtual functions.

**Table 31-3. PCIe EP Function Mapping**

| EP | Functional Blocks | PCIe* Function Number | BARs | Interrupt Capability | PCIe Extended/PCI Capabilities |
|---|---|---|---|---|---|
| Intel® QuickAssist Technology | Intel® QAT | 0 | PMISCBAR, PETRINGCSRBAR | MSI, MSI-X with 16 vectors, INTA | MSI, MSI-X, PM, PCIe, AER, SR-IOV, ARI |
| VF[1:16] | | 8-23 Note 1 | SRIOVBAR0, SRIOVBAR1 | MSI only | MSI, PCIe, AER, ARI, ACS |

***Notes:***
1. ARI must be enabled to view function numbers > 7.
2. When ARI is disabled, the bus and device number is captured by the RI on every type 0 configuration write. When ARI is enabled, the device number is implicitly assumed to be 0 and only the bus number is captured by RI on every type 0 configuration write.
3. PPETRINGCSRBAR is used for access to registers that configure and access Ring. PMISCBAR is used for accessing EP and the MSI-X table.

### 31.3.4.5 EP Mapping of BARs to MMIO

Table 31-4, "EP Mapping of BARs to MMIOs" shows the mapping of the EP MMIO regions to the BARs. See each PCIe function BAR definition for more details.

**Table 31-4. EP Mapping of BARs to MMIOs**

| Function # | BAR Name (Size) | Region Size | Functionality | Comments |
|---|---|---|---|---|
| 0 | PMISCBAR (256 KB) | 248 KB | EP, Intel® QAT | This region is mapped to the EP and Intel® QAT CSRs. |
| 0 | | 4 KB | EP MMIO CSRs | The EP CSRs are located in the EP. |
| 0 | | 4 KB | MSI-X Tables | The MSI-X tables are located in the EP. |
| 0 | PETRINGCSRBAR | 256 KB | 16 Bundles of Ring CSRs | 128 KB for all 16 bundles of CSRs and Intel® QAT CSRs. ***Note:*** This space is allocated for 32 bundles. Only the lower 16 bundles are used and the upper 16 are memory holes in this region. |
| 8-23 (VF[1-16]) | SRIOVBAR0 | MIN (4 KB, System Page Size) (Note_2) | 1 ETR Bundle per VF[x] (RAM) | Each bundle requires 512B, but are mapped in a 4 KB window by default or the system page size. |
| | SRIOVBAR1 | MIN (4 KB, System Page Size) (Note_3) | VF Messaging Registers | There is only one register per VF but is mapped in a 4 KB window by default or the system page size. |

***Notes:***
1. Accesses to the MMIO region that are not implemented within the regions defined in this table will return 0s. Accesses to MMIO regions that are not claimed by any of the BARs will return UR.
2. The SRIOVBAR0 size is dependent on the value programmed in the system.
3. The SRIOVBAR1 size is dependent on the value programmed in the system page size.

### 31.3.4.6 Alternative Routing-ID Interpretation (ARI)

Receive Interface (RI) provides support for the Alternative Routing-ID Interpretation (ARI). When enabled, ARI allows the PCIe device to implement more than 8 functions.

An ARI device interprets the Routing, Requester, and Completer IDs as having an 8-bit function number instead of the traditional 3-bit function number. An ARI device has its associated device number implied to be 0 rather than specified by an ID field.

*Note:* ARI must be enabled to use all of the VF Intel® QAT supports.

**Table 31-5. Bus, Device and Function Interpretation**

| | | | |
|---|---|---|---|
| **ARI Disabled** | **Bus Number** | **Device Number** | **Function Number** |
| | [7:0] | [4:0] | [2:0] |
| **ARI Enabled** | **Bus Number** | **Function Number** | |
| | [7:0] | [7:0] | |

RI supports ARI for function 0 when SR-IOV is enabled.

### 31.3.5 Interrupts

RI is responsible for collating interrupts and errors from the endpoint and taking the appropriate actions based on the PCIe configurations status.

Interrupts from EP can be configured in the RI to be routed to the IA cores. The interrupts can be sent upstream as either an MSI, MSI-X, or by using INTx virtual wire interrupt signaling mechanism (which are messages on PCIe).

#### 31.3.5.1 INTx

PCIe supports devices that need to use the PCI compatible legacy interrupt signaling: INT[A, B, C, D]. Instead of dedicated pins for this functionality, PCIe supports inband virtual wires.

The EP supports the generation of INTx virtual wire interrupt signaling based on internal generated interrupts. The INTx notification is accomplished using PCIe messages.

RI is responsible for collating interrupt sources from the EP agents and generating the transaction that is translated to a PCIe interrupt message to the IA. Upon detecting an interrupt from an internal source, the RI first determines whether to generate an MSI, MSI-X, or INTx message.

#### 31.3.5.2 Message Signaled Interrupt (MSI)

Message Signaled Interrupts (MSI) are an optional feature that enables a device function to request service by writing a system-specified data value to a system-specified address (using a PCI Dword memory write transaction). System software initializes the message address and message data during device configuration. Each function in the EP supports the MSI capability. All the interrupts per function share one vector.

#### 31.3.5.3 MSI-X

MSI-X defines a separate optional extension to the basic MSI functionality. Compared to MSI, MSI-X supports a larger number of vectors per function, the ability for the OS to program independent address and data values for each vector. The address and data values for each vector are specified by a table that resides in the memory space. MSI-X is supported only for the physical function.

Once an MSI has been generated using MSI-X, its characteristics are similar to an MSI that was generated using an MSI capability record. However, MSI-X supports per-vector masking that is optional for MSI. MSI-X also supports a Function Mask bit, which when set, masks all of the vectors associated with a function. Per-vector masking is managed through a Mask and Pending bit pair per the MSI vector or MSI-X table entry. An MSI-X vector is masked when its associated MSI-X Table Entry Mask bit or the MSI-X Function Mask bit is set. While a vector is masked, the function is prohibited from sending the associated message, and the function must set the associated Pending bit whenever the function would otherwise send the message. When software unmasks a vector whose associated Pending bit is set, the function must schedule sending the associated message, and clear the Pending bit as soon as the message has been sent.

When MSI-X is enabled, EP generates an interrupt using a PCIe posted memory write transaction. The address and data of that transaction are determined by the system and programmed in the PCI MSI-X table entry.

### 31.3.6 EP Errors

There are two classes of errors that the EP will encounter:

- Errors that occur due to PCIe transactions
  - PCIe error management defines the scope of these kinds of errors including how they are reported and handled.
- Other errors not directly related to PCIe transactions
  - This class of errors is reported via interrupts and does not fall under the scope of PCIe error management mechanisms.

#### 31.3.6.1 PCIe Error Management

PCIe error management focuses on errors associated with the PCIe interface and the transactions between the transaction layers of the transmitting and the receiving PCIe ports.

#### 31.3.6.2 PCIe Error Reporting Mechanisms

The PCIe specification provides three mechanisms for reporting errors that occur while servicing PCIe transactions.

1. PCI
   a. This mechanism provides backward compatibility with legacy PCI compatible software and is required per the PCIe specification. Legacy compatible PCI software will use this mechanism and control this feature via the PCI configuration Command register. When this mechanism is enabled, EP will also log error status information in the PCI Configuration Status register.
   b. EP will support the legacy PCI compatible error mechanism.
2. PCIe
   a. This is the baseline PCIe mechanism for software that understands PCIe devices. The mechanism can be enabled via the PCIe Device Control register. Error status is logged in the PCIe Device Status register.
   b. Supported error classifications: unsupported request type, fatal error, non-fatal error, and correctable error
3. PCIe Advanced Error Reporting (AER)
   a. The optional advanced error reporting registers can be implemented by PCIe devices. This mechanism is defined using a PCIe extended capability structure.
   b. All EP Functions will support AER.

### 31.3.6.3 PCIe Error Handling and Signaling

The EP will classify errors in the following three PCIe specification-defined buckets:

1. Correctable errors: These errors are handled by the Hardware (HW).
2. Uncorrectable errors (non-fatal): These errors are handled by device-specific software.
3. Uncorrectable errors (fatal): These errors are handled by system software.

EP will use three different mechanisms to signal errors that occur when processing PCIe commands.

1. Completion status
2. Error forwarding or data poisoning
3. Error messages

### 31.3.6.4 PCIe Error Sources

Table 31-6, "PCIe Error Sources" defines the standard PCIe defined error sources. The following sections describe each of the PCIe error types, and describe how the EP handles the PCIe errors.

**Table 31-6. PCIe Error Sources**

| | Type of Error | Reported Using | Comment |
|---|---|---|---|
| TL | ECRC Check | Uncorrectable Non-Fatal Message | |
| | Malformed TLP | Uncorrectable Fatal Message | RI will perform checks on the PCIe* transactions received from IA. |
| | Completion Timeout | Uncorrectable Non-Fatal Message | RI will generate these split transaction errors. Completer abort is supported in RI. |
| | Unsupported Requests | Completion Status | |
| | Completer Abort (optional) | | |
| | Unexpected Completion | Uncorrectable Non-Fatal Message | |
| | Data Corruption/Poisoning | PCIe Header | TI will set the EP bit in the header based on the internal bus data error.<br>Writes that terminate in RI get dropped and the status logged.<br>Writes that target an EP function will be sent with a data error and logged in RI. |

#### 31.3.6.4.1 Role-Based Error Reporting

In earlier versions of the PCI Express Specification, errors were reported by the agent that detected the error. The *PCI Express Base Specification Revision 3.0* implements a role-based error reporting where the response to the errors is based on the component's role in the transaction. In general, errors detected in non-posted transactions are handled by the initial requester and the completer may optionally send an advisory message to the root complex as an ERR_COR message. Errors in posted transactions are still logged and reported by the target device.

*Note:* If the severity for the error is programmed to fatal in the PCI Express Uncorrectable Error Severity register, then it is not an advisory non-fatal error and is signaled with an ERR_FATAL message. A fatal severity overrides all other Advisory Error Control bits.

The following errors are considered advisory non-fatal error cases and have different handling depending on the transaction type.

- ECRC check failed
- Unsupported Request (UR)
- Completer Abort (CA)
- Unexpected completion
- Poisoned TLP received
- Completion timeout

#### 31.3.6.4.2 Malformed Packets

The following checks are made to detect malformed TLPs.

- Data payload exceeds the length specified by the value in the Max_Payload_Size field of the Device Control register.

When a malformed packet is detected, the packet is dropped and the error is logged. No flow control information is updated for malformed packets.

#### 31.3.6.4.3 ECRC Check Failed

EP does not calculate for ECRC but will report the ecrc_error signal that it receives. EP reports by sending an ERR_NONFATAL message to the root complex.

#### 31.3.6.4.4 Unsupported Request

Unsupported requests are detected by the address decode and translation logic. A TLP is treated as unsupported in the following cases:

- The TLP fails to match any of the active memory or I/O windows.
- A configuration TLP that targets an invalid function number.
- A receipt of a Vendor_Defined Type 0 message.
- A message request with an undefined or unsupported message code.
- A poisoned I/O or configuration request.
- A receipt of a memory or I/O transaction while in a non-D0 power state.
- A receipt of a Memory Read Lock (MRdLk).

No checks are made for the address plus the length crossing a window boundary.

For posted transactions, this is **not** an advisory error, and an ERR_NONFATAL is sent to the root complex.

For non-posted transactions, this is considered an advisory error. An ERR_COR is sent to the root complex and a completion with UR status is returned to the requester.

*Note:* If the severity setting in the PCI Express Uncorrectable Error Severity register is fatal, this is not an advisory error, and an ERR_FATAL will be sent to the root complex.

#### 31.3.6.4.5 Completer Abort (CA)

Requests that target abort or master abort on the internal bus are treated as a completer abort.

These requests must have passed the malformed TLP checks as well as the unsupported request checks before they are issued on the internal bus.

For posted transactions, an ERR_NONFATAL is sent to the root complex.

For non-posted transactions, an ERR_COR is sent to the root complex and a completion with CA status is returned to the requester.

*Note:* If the severity setting in the PCI Express Uncorrectable Error Severity register is fatal this is not an advisory error and an ERR_FATAL will be sent to the root complex.

#### 31.3.6.4.6 Unexpected Completions

Unexpected completions occur when a completion transaction ID does not match a outstanding request. If the Requester ID of the completion matches a valid function, the error will get logged in that function. Otherwise, the error will get logged against all functions.

This is an advisory non-fatal error and an ERR_COR will be sent to the root complex.

*Note:* If the severity setting in the PCI Express Uncorrectable Error Severity register is fatal, this is not an advisory error and an ERR_FATAL will be sent to the root complex.

#### 31.3.6.4.7 Poisoned TLP Received

Poisoned TLPs can be received for both inbound posted (write/message) and inbound completions. The two TLP types can be handled differently.

**Poisoned completions** are passed through to the target agent with the error bit set. The error is logged in the corresponding function. This is an advisory error and an ERR_COR will be issued.

**Poisoned memory writes** that are MMIO-mapped are passed through to the intended target. Depending on the register target, the MMIO-mapped register may get updated with the poisoned data, whereas in some cases the register will not be updated as the byte-enable signals will be disabled. The error is logged in the corresponding function. This is a non-fatal error and an ERR_NONFATAL will be issued.

Writes to configuration registers will not complete to the target. Since configuration writes are non-posted writes, a completion with UR is returned to the host.

*Note:* If the severity setting in the PCI Express Uncorrectable Error Severity register is fatal, this is not a non-fatal error, and an ERR_FATAL will be sent to the root complex. Auto-recovery is discouraged as the ERR_FATAL message will likely bring down the hierarchy.

#### 31.3.6.4.8 Completion Timeout

When an out-bound, non-posted request results in a completion timeout, the advanced error registers are updated in the corresponding function. This will be treated as an advisory error, and an ERR_COR will be sent to the root complex.

*Note:* If the severity setting in the PCI Express Uncorrectable Error Severity register is fatal, this is not an advisory error, and an ERR_FATAL will be sent to the root complex. Auto-recovery is discouraged as the ERR_FATAL message will likely bring down the hierarchy.

#### 31.3.6.4.9 Non-Function Specific Errors

The PCI Express Specification lists the following errors as non-function specific:

- Transaction layer errors
  - — ECRC fail
  - — UR, when caused by no function claiming a TLP
  - — Receiver overflow
  - — Flow control protocol error
  - — Malformed TLP
  - — Unexpected completion, when caused by a no function claiming a completion

On the detection of one of these errors, a multi-function device should generate at most one error reporting message of a given severity, where the message must report the Requester ID of a function of the device that is enabled to report that specific type of error. If no function is enabled to send a reporting message, the device does not send a reporting message. If all reporting-enabled functions have the same severity level set for the error, only one error message is sent. If all reporting-enabled functions do not have the same severity level set for the error, one error message for each severity level is sent. Software is responsible for scanning all functions in a multi-function device when it detects one of those errors.

### 31.3.7 Device-Specific Error Management

EP will report device-specific errors using interrupts—INTx or MSI based on the PCIe configuration settings.

**Table 31-7. Device-Specific Errors**

| Function | Description |
|---|---|
| Function 0 | EP Implements error source and mask registers.<br>When detecting an assertion on an unmasked error, RI will generate either an INTA or MSI. |

#### 31.3.7.1 Memory Error Poisoning

The TI also tracks data errors for all transactions that flow through the TI, including accesses to PCIe and CSRs.

##### 31.3.7.1.1 Memory Write Poisoning

For a PCIe memory write (including memory writes triggered by the Ring Controller), the TI pulls data from the EP master. If the internal bus indicates an error, then the TI logs the error, but it does not abort the write transaction. Instead, the TI passes along the data error notification along with the data. On PCIe, the TI will set the EP bit in the header of the packet.

### 31.3.8 Error Handling with SR-IOV

EP ring-related errors that are associated with a given VF are handled by the given VF itself. For example, the VF would report and log the errors. Each VF supports the legacy PCI (PCI Command and Status registers) and PCIe reporting mechanism (ERR_COR, ERR_NONFATAL, ERR_FATAL messages). The VF also supports PCIe extended capability - AER. Errors that are not directly related to the ring associated with a given VF are handled by the PF. For example, errors such as EP parity errors, etc.

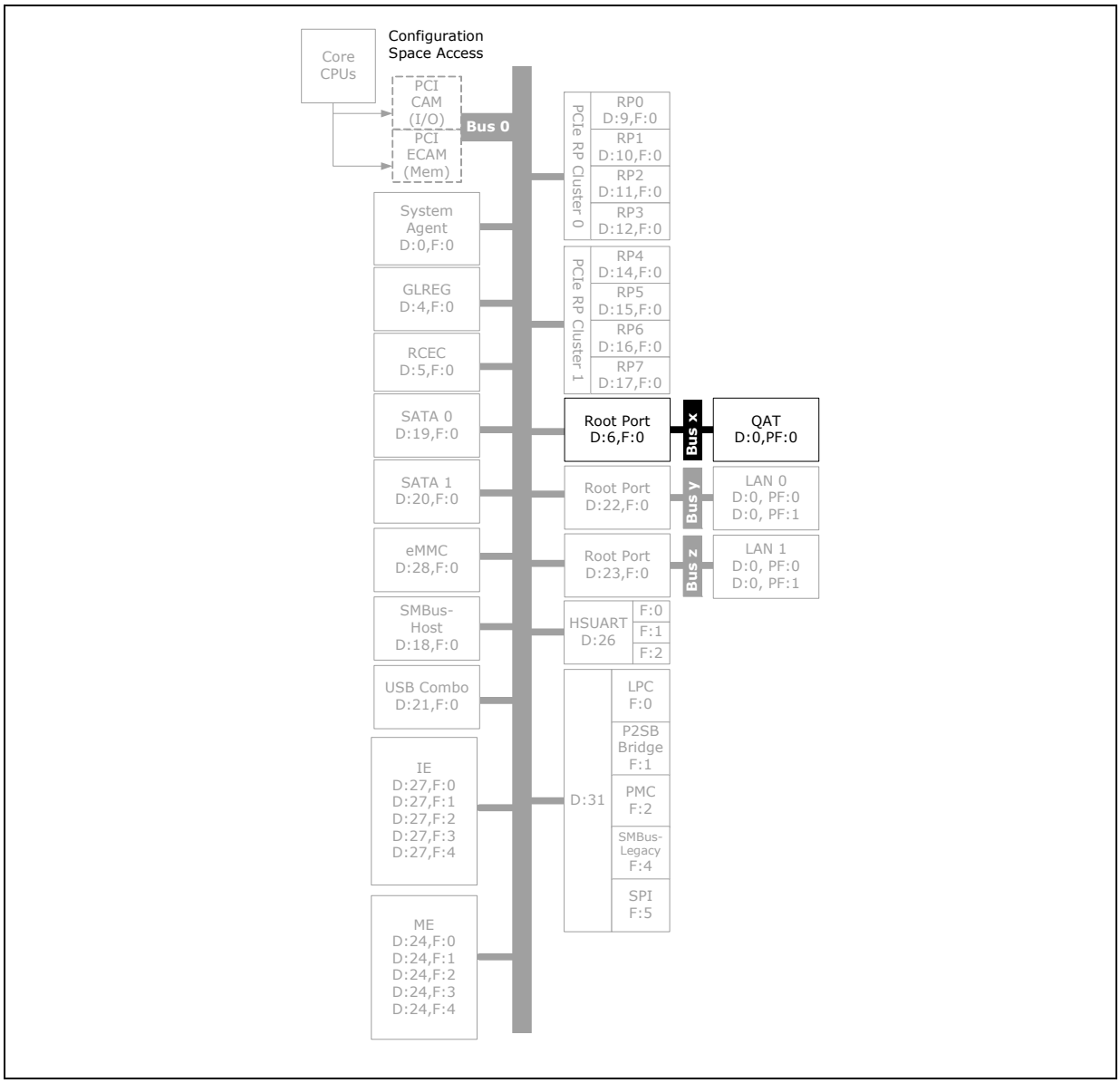
## 31.4 PCI Configuration Space Map

Figure 31-5, "Register Map" shows the associated PCI configuration space from a system software viewpoint. Refer to the *Intel® QuickAssist Technology API Programmer's Guide* for information on the Intel® QuickAssist Technology APIs to program the Intel® QAT available at:

- https://01.org/packet-processing/intel%C2%AE-quickassist-technology-drivers-and-patches.

The APIs communicate with the Intel® QuickAssist Technology hardware via PCI configuration space access and assisted rings stored in system memory.

**Figure 31-5. Register Map**




§ §

# 32 SoC Voltage Rails

This chapter lists and describes all of the power supplies that directly connect to the SoC, as well as signals that directly connect to the SoC and are related to the power supplies.

This chapter does NOT discuss the complete set of platform (the printed wiring board containing the SoC as well as all other associated components) power supplies, those that do not connect to the SoC. A complete platform requires additional voltages, beyond what is directly connected to the SoC.

## 32.1 SoC Voltage (And Related Signals) Descriptions

The signal descriptions are shown in Table 32-1, "Signal Descriptions." The "Direction" column of Table 32-1, "Signal Descriptions" is interpreted as follows:

- I = Input signal.
- O = Output signal.

The "Voltage Regulator Group" column of Table 32-1, "Signal Descriptions" is used as an aid in categorizing the signal type; implying a particular voltage supply level, class of voltage supply, if the signal is a reference voltage, if the signal is used for Voltage Regulator (VR) sensing (SENSE), Observation (OBS), or a VR return (VSS).

Two signals that are of the same "Voltage Regulator Group" should not necessarily be directly interconnected. While many signals are supplied by the same VR, they may have differing specific filtering and/or isolation requirements. For example, a signal may require a ferrite bead, an inductor, or some other R/C filtering circuit. The Platform Design Guide should be referenced when determining how to interconnect signals for a successful platform design.

For pin numbers and other information for these signals and voltage sources, see Table 39-5, "Package Rotation (ROT) Pins" and Table 39-6, "Directory of Signal and Voltage Pins - Sorted by Pin Number."

## Table 32-1. Signal Descriptions (Sheet 1 of 2)

| Signal Name | Direction | Voltage Regulator Group | Description |
|---|---|---|---|
| OBS_VCCCPUVIDSI0GT_MOD1 | O | OBS | Used to observe the die voltage being delivered to Atom CPU core #0. |
| OBS_VCCCPUVIDSI0GT_MOD4 | O | OBS | Used to observe the die voltage being delivered to Atom CPU core #1. |
| OBS_VCCBIUVIDSI0GT_MOD1 | O | OBS | Used to observe the die voltage being delivered to the BIU. |
| OBS_VCCRAMCPUSI1GT_ MOD0 | O | OBS | Used to observe the die voltage being delivered to the Atom CPU core L2 caches. |
| SENSE_VCCCPUVIDSI0 | O | SENSE | Voltage sense signal that provides feedback to the voltage regulator that supplies the voltage to the Atom CPU cores via the VCCCPU_SVID_S0 set of pins. |
| SENSE_VSSCPUVIDSI0 | O | SENSE | Voltage sense return signal for the voltage sense signal SENSE_VCCCPUVIDSI0 that provides feedback to the voltage regulator that supplies the voltage to the Atom CPU cores via the VCCCPU_SVID_S0 set of pins. |
| SENSE_VSSRAMCPU_S0 | O | SENSE | Voltage sense return signal for the voltage sense signal SENSE_VCCRAMCPU_S0 that provides feedback to the voltage regulator that supplies the voltage to the Atom CPU core L2 caches via the VCCRAM_S0 set of pins. |
| SENSE_VCCRAMCPU_S0 | O | SENSE | Voltage sense signal that provides feedback to the voltage regulator that supplies the voltage to the Atom CPU core L2 caches via the VCCRAM_S0 set of pins. |
| VCCFHVSIO_1P05 | I | V1P05 | Power supply for an on-die voltage regulator. |
| VCCACKDDR_1P05 | I | V1P05 | Power supply pins for the DDR digital clock. |
| VCCADDR_1P05 | I | V1P05 | Power supply pins for the DDR I/O. |
| VCCAON_HSIO_1P05 | I | V1P05 | Power supply pins that feeds the logic which controls the LDO/PG within the HSIO PCIe3, SATA 3.1, and USB3 PHYs. |
| VCCCL_HSIO_1P05 | I | V1P05 | Power supply pins that power the PLLs within the HSIO PCIe3, SATA 3.1, and USB3 PHYs. |
| VCCDIGICKSI0_1P05 | I | V1P05 | Power supply pins for the SoC internal integrated digital clock circuit, requires on-board filtering. |
| VCCDIGXXXSI0_1P05 | I | V1P05 | Power supply pins for multiple digital IP block. |
| VCCFHVSI0_GROUP0_1P05 | I | V1P05 | Power supply for an on-die voltage regulator. |
| VCCFHVSI0_GROUP1_1P05 | I | V1P05 | Power supply for an on-die voltage regulator. |
| VCCFHVSI0_GROUP2_1P05 | I | V1P05 | Power supply for an on-die voltage regulator. |
| VCCKRA_LV | I | V1P05 | Power supply pins feeding 10GE IP block. |
| VCCKRLCPLL | I | V1P05 | Power supply pins for the 10GE PLL, requires on-board filtering. |
| VCCPLLEBB_HSIO_1P05 | I | V1P05 | Power supply pin that powers T-Line drivers within the HSIO PHYs. |
| VCCUSB_1P05 | I | V1P05 | Power supply pins that power the USB2 PHY. |
| VCCFHVIFPSI0_1P8 | I | V1P80 | Power supply for an on-die voltage regulator. |
| VCCPADXXXSI0_1P8 | I | V1P80 | Power supply pins that power the 1.8V GPIOs. |

## Table 32-1. Signal Descriptions (Sheet 2 of 2)

| Signal Name | Direction | Voltage Regulator Group | Description |
|---|---|---|---|
| VCCUSBSUS_1P8 | I | V1P80 | Power supply pins that power the 1.8V suspend supply for the USB2 PHY. |
| VCCFHVIFPSI0_3P3 | I | V3P30 | Power supply for an on-die voltage regulator. |
| VCCPADXXXSI0_3P3 | I | V3P30 | Power supply pins that power the 3.3V GPIOs, and RTC. |
| VCCUSBSUS_3P3 | I | V3P30 | Power supply pins that power the 3.3V suspend supply for the USB2 PHY. |
| VCC_LPC_ESPI_3P3_1P8 | I | V3P30 | Power supply for the LPC 3.3V. |
| VCCCPU_SVID_S0 | I | VCC | Power supply pins that power the Atom cores, actual voltage is dependent on SoC request to the VR. |
| VCCRAM_S0 | I | VCCRAM | Power supply pins that power the Atom L2 caches, actual voltage is dependent on SoC request to the VR. |
| VCCKR_HV | I | V1P80 | Voltage reference used by the LAN IP block, requires on-board filtering. |
| VCCREF_SFRXXXSI0_N | I | VCCREF | Voltage reference for the North side Atom core LDOs, requires on-board filtering. |
| VCCREF_SFRXXXSI0_S | I | VCCREF | Voltage reference for the South side Atom core LDOs, requires on-board filtering. |
| VCCUSB_1P24 | I | VCCREF | Voltage reference for the USB2 SFR LDO. |
| VCCCKDDR_VDDQ | I | VDDQ | Power supply pins that drive the DDR clock signals to the DDR memory devices, actual voltage is dependent on SoC request to the VR. |
| VCCDDR_VDDQ | I | VDDQ | Power supply pins that drive the DDR data signals to the DDR memory devices, actual voltage is dependent on SoC request to the VR. |
| VCCDDRSFR_VDDQ | I | VDDQ | Power supply for the SFR LDO feeding the DDR PLL, requires on-board filtering, actual voltage is dependent on SoC request to the VR. |
| VNN_SVID | I | VNN | Power supply pins that power portions of several IPs and level shifters within the UnCore, actual voltage is dependent on SoC request to the VR |
| VCCRTC_3P3_G3 | I | VRTC | Battery backed power supply for the RTC IP block. |
| VSS | O | VSS | SoC common power supply return signal.<br>Check the Platform Design Guide to see if any of the VSS pins have unique routing or isolation requirements. |

§ §

# 33 Resets and Voltage Sequencing

This chapter discusses the requirements for resetting the SoC and the power-up/power-down voltage sequencing requirements for the SoC. This chapter discusses only the requirements of the SoC, not the entire platform (Printed Wiring Board (PWB) containing the SoC and all associated components).

**Table 33-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| *ACPI Specification* | 5.0 A | November 2013 | *Advanced Configuration and Power Interface Specification Version 5.0, Errata A* |

## 33.1 Signal Descriptions

The signal descriptions are shown in Table 33-2, "Signal Names and Descriptions."

For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing."

The Direction/Type column of Table 33-2, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.

**Table 33-2. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| PMU_PLTRST_N | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| PMU_SLP_S3_N | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| PMU_SLP_S45_N | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| PMU_SUSCLK | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| SUS_STAT_N | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| SUSPWRDNACK | O | Yes | See Chapter 16, "Power Management Controller (PMC)" for details. |
| COREPWROK | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| RSMRST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| RTEST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |
| SRTCRST_N | I | No | See Chapter 24, "Real Time Clock (RTC)" for details. |

Nine separate power supplies are required to power the SoC. This chapter does not detail how the nine power supplies connect to the SoC, see Chapter 32, "SoC Voltage Rails."

For convenience, Table 33-3, "SoC Required Power Supplies" lists the reference names of the SoC required power supplies, and their nominal voltages:

**Table 33-3. SoC Required Power Supplies**

| PS Name | Nominal Voltage (V) | Comment |
|---|---|---|
| VRTC | 2.0-3.3 | Tied to battery or 3.3V |
| VCC | 0.52-1.24 | Dynamic Range (SVID address 00h) |
| VNN | 0.65-1.24 | Dynamic Range (SVID address 01h) |
| VDDQ | 1.20 | DDR4 (SVID address 02h) |
| VCCRAM | 0.75-1.20 | Dynamic Range (SVID address 03h) |
| VCCREF | 1.24 | Static (SVID address 015h) |
| V1P05 | 1.05 | Static (SVID address 015h) |
| V1P80 | 1.80 | Static (SVID address 015h) |
| V3P30 | 3.3 | Static (SVID address 015h) |

Table 33-4, "SoC Power Supply for each Supported Power State" lists the states of each SoC power supply for each supported power state:

**Table 33-4. SoC Power Supply for each Supported Power State**

| PS Name | G3 | S5 | S4 | S0 |
|---|---|---|---|---|
| VRTC | With Battery: On<br>Without Battery: OFF | ON | ON | ON |
| VCC | OFF | OFF | OFF | ON |
| VNN | OFF | ON | ON | ON |
| VDDQ | OFF | OFF | OFF | ON |
| VCCRAM | OFF | OFF | OFF | ON |
| VCCREF | OFF | OFF | OFF | ON |
| V1P05 | OFF | ON | ON | ON |
| V1P80 | OFF | ON | ON | ON |
| V3P30 | OFF | ON | ON | ON |

## 33.2 Architectural Overview

The SoC device configuration and power management complies with the *Advanced Configuration and Power Interface Specification Version 5.0, Errata A*. However, not all the states defined by ACPI are supported.

The supported ACPI Global (G) States are listed below:

- G0
- G1
- G2
- G3

The supported ACPI System (S) States are listed below:

- S0
- S4
- S5

ACPI terms are used in this chapter for the SoC Reset and Voltage Sequencing requirements.

Table 33-5, "Supported ACPI Transitions And Definitions" lists the supported ACPI Power State transitions along with the term used throughout this chapter in describing each event:

**Table 33-5. Supported ACPI Transitions And Definitions**

| ACPI Transition | SoC |
|---|---|
| From G3 To G0.S0 | Cold Boot |
| From G0.S0 To G3 | Shutdown |
| From G0.S0 To G1.S4 | Deep Sleep |
| From G1.S4 To G0.S0 | Deep Sleep Wake |
| From G0.S0 To G2.S5 | Soft Off |
| From G2.S5 To G0.S0 | Soft Off Wake |
| From Gx.Sx To G3 | Power Failure |

The sequencing/reset requirements for each ACPI Transition listed in Table 33-5 is detailed in the sections below.

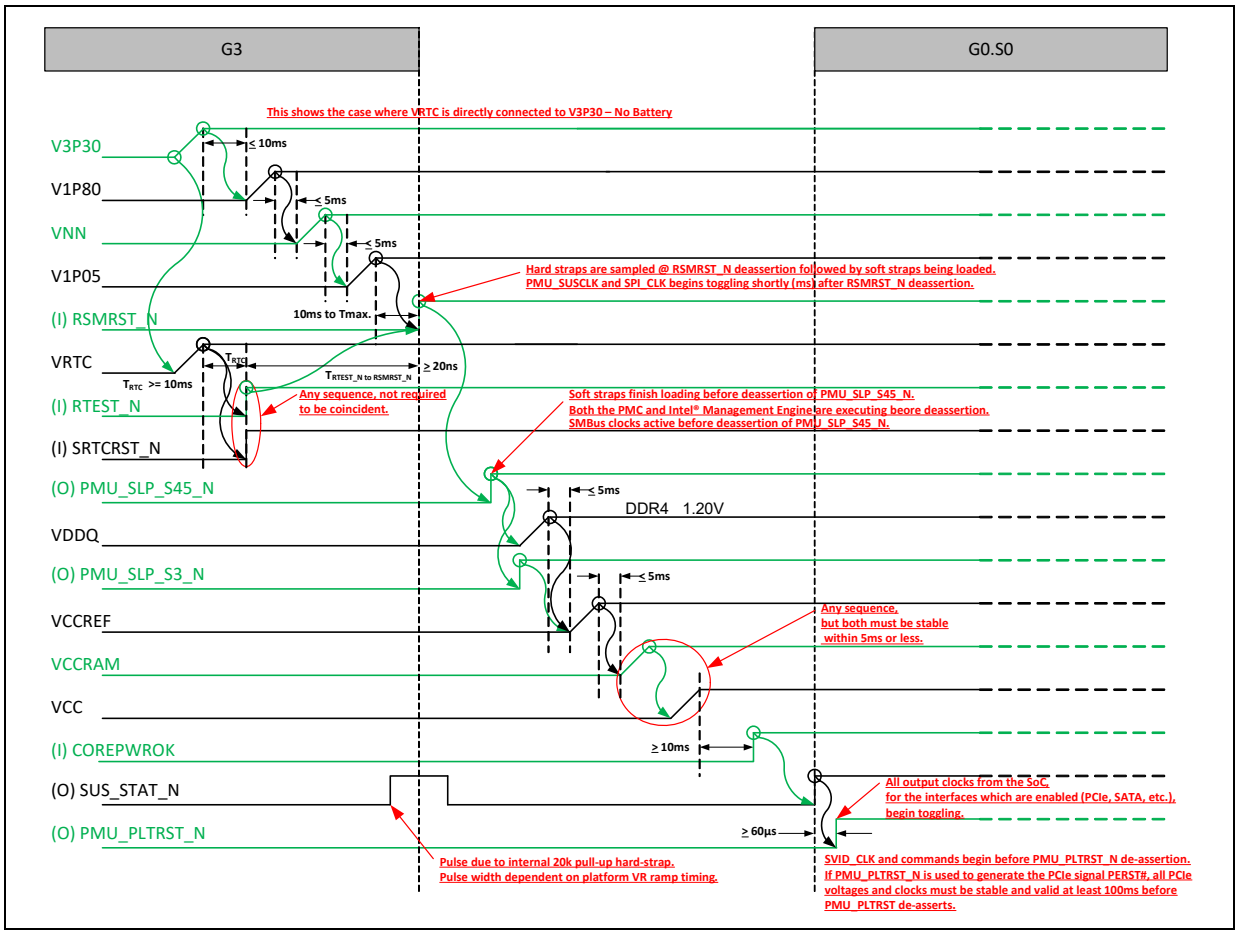
## 33.3 Cold Boot: From G3 To G0.S0

Figure 33-1, "ACPI Cold Boot Sequence" illustrates the ACPI timing sequence in going from G3 to G0.S0. The arrows in the timing diagram show the sequences that must be followed. In some cases a delay is shown, in other cases, no specific timing delay is given. For cases where no specific timing delay is given, only the sequence is of significance.

The arrows showing the transition of one signal/voltage causing the transition of another signal/voltage occur when the signal/voltage causing the transition reaches its specified valid level. For example, in Figure 33-1, "ACPI Cold Boot Sequence," when V3P30 reaches the level specified as the operating range for V3P30, this is when V1P80 can begin to reach its specified valid level. And for this example, V1P80 must begin ramping within 10ms or less after V3P30 has reached its valid operating range.

Figure 33-2, "Non-ACPI Cold Boot Sequence" is similar to Figure 33-1, "ACPI Cold Boot Sequence," but illustrates the Non-ACPI sequence, where the PMU_SLP_S45_N and PMU_SLP_S3_N signals may be ignored.

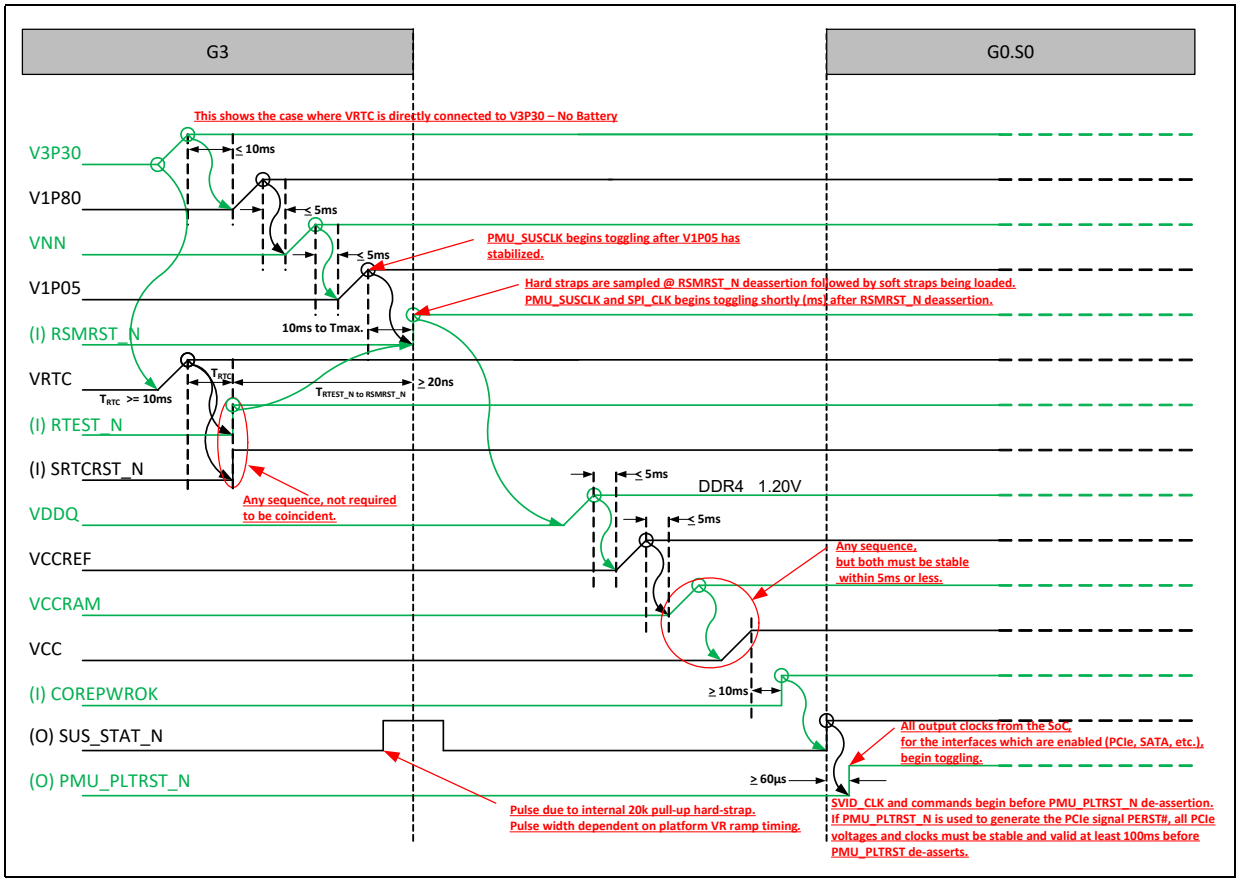
**Figure 33-1. ACPI Cold Boot Sequence**




***Notes:***

1. After RSMRST_N de-asserts the Hard straps are sampled. PMU_SLP_S45 will not de-assert until PMU_SUSCLK starts and the Soft straps are loaded from flash.
2. Before PMU_SLP_S45_N de-asserts the PMC and Intel® Management Engine are running.
3. TMAX = 10s: RSMRST should not be asserted indefinitely while the uncore supplies are powered. Specifically, a maximum delay of 10s from V1P05 to RSMRST deassertion should be enforced.

## Figure 33-2. Non-ACPI Cold Boot Sequence




***Notes:***
1. After RSMRST_N de-asserts the Hard straps are sampled. PMU_SLP_S45 will not de-assert until PMU_SUSCLK starts and the Soft straps are loaded from flash.
2. Before PMU_SLP_S45_N de-asserts the PMC and Intel® Management Engine are running.
3. TMAX = 10s: RSMRST should not be asserted indefinitely while the uncore supplies are powered. Specifically, a maximum delay of 10s from V1P05 to RSMRST deassertion should be enforced.

Figure 33-1, Figure 33-2, Figure 33-3, and Figure 33-4 show the configuration where there is no battery connected to VRTC, and instead optionally connected only to V3P30. However, if the battery backed option is selected, then VRTC would always be at the battery voltage and ramp up to V3P30 after the V3P30 VR comes up. When the battery backed option is selected, then the battery and V3P30 are both connected to the VRTC pin through diodes.

The signals RTEST_N and SRTCRST_N are normally held high to VRTC through an external RC circuit, causing a delay in the signals becoming inactive sometime after VRTC has ramped. On a typical system, a jumper is used to force these signals active for testing purposes.

*Note:* In certain user cases, the core power rails (VCC/VCCRAM/VCCREF/VDDQ) do not need to be cycled during cold reset, but COREPWROK must be cycled during cold reset. If this certain user cases is using non-ACPI code boot sequence, SLP_Sx still are signals to be monitored to be de-asserted before COREPWROK asserted.

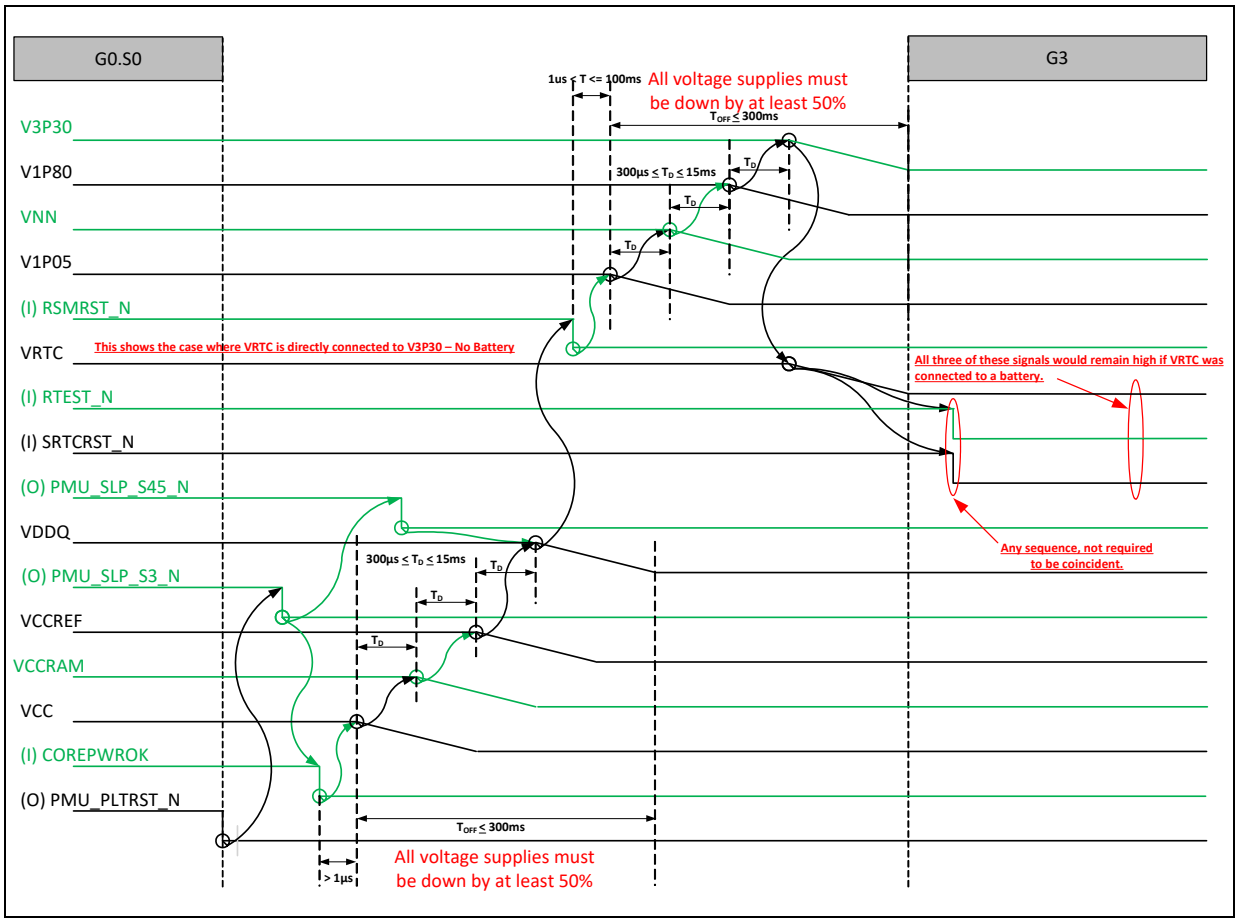
## 33.4 Shutdown: From G0.S0 To G3

Figure 33-3, "ACPI Shutdown Sequence" illustrates the ACPI timing sequence in going from G0.S0 to G3. The arrows in the timing diagram show the sequences that must be followed. In some cases a delay is shown, in other cases, no specific timing delay is given. For cases where no specific timing delay is given, only the sequence is of significance.

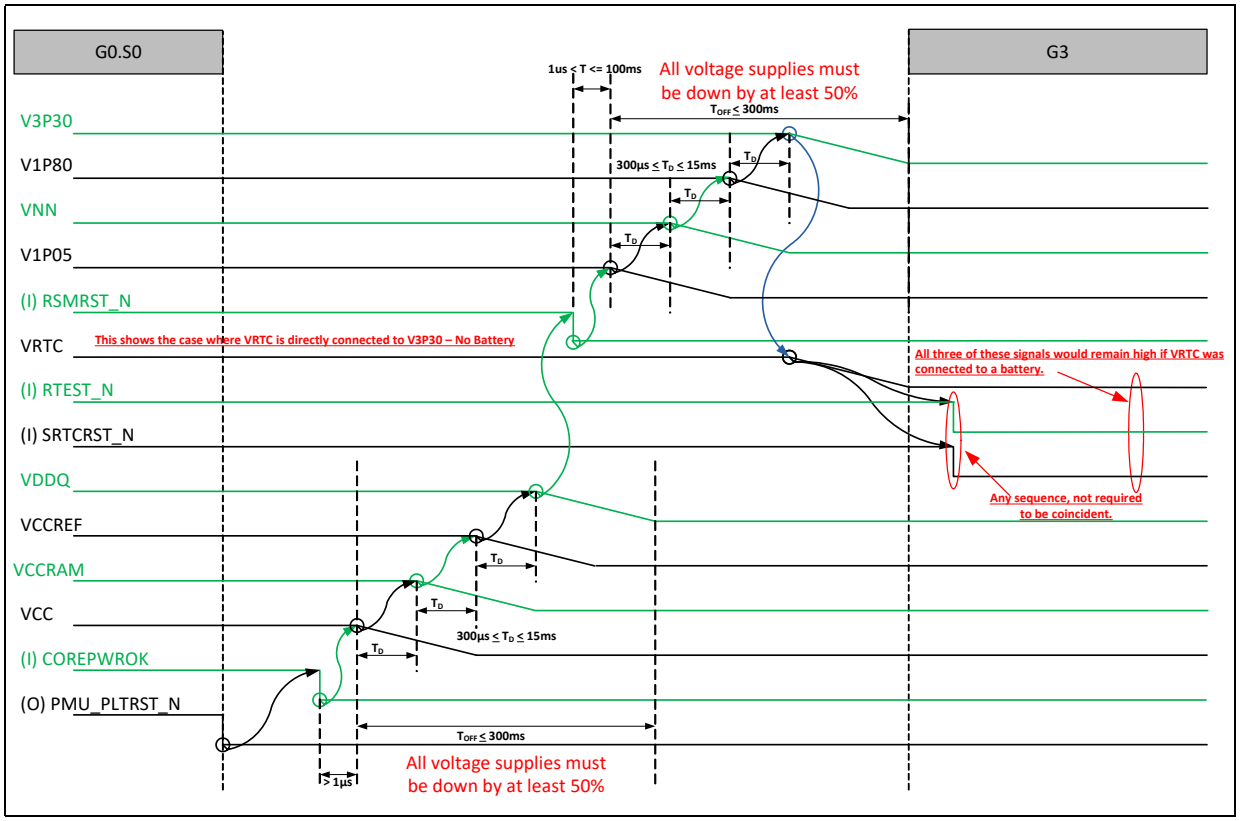
The arrows showing the transition of one signal/voltage causing the transition of another signal/voltage occur when the signal/voltage causing the transition falls out of its specified valid level. For example, in Figure 33-3, "ACPI Shutdown Sequence," when V1P05 falls below its valid operating level, this is when the timer begins for $T_D$. Where $T_D$ is defined as the required time delay before VNN can begin ramping down.

Figure 33-4, "Non-ACPI Shutdown Sequence" is similar to Figure 33-3, "ACPI Shutdown Sequence," but illustrates the Non-ACPI sequence, where the PMU_SLP_S45_N and PMU_SLP_S3_N signals may be ignored.

**Figure 33-3. ACPI Shutdown Sequence**



## Figure 33-4. Non-ACPI Shutdown Sequence



## 33.5 Deep Sleep: From G0.S0 To G1.S4

Going from G0.S0 to G1.S4 is depicted by the sequence shown in Figure 33-3, "ACPI Shutdown Sequence," and stopping after the VDDQ power supply is turned off. There is no way to discern the G1.S4 state from the G2.S5 state from merely observing the SoC signals.

## 33.6 Deep Sleep Wake: From G1.S4 To G0.S0

Going from G1.S4 to G0.S0 to is depicted by the sequence shown in Figure 33-1, "ACPI Cold Boot Sequence," beginning in the G1.S4 state. There is no way to discern the G1.S4 state from the G2.S5 state from merely observing the SoC signals.

## 33.7 Soft Off: From G0.S0 To G2.S5

Going from G0.S0 to G2.S5 is depicted by the sequence shown in Figure 33-3, "ACPI Shutdown Sequence," and stopping after the VDDQ power supply is turned off. There is no way to discern the G1.S4 state from the G2.S5 state from merely observing the SoC signals.

## 33.8 Soft Off Wake: From G2.S5 To G0.S0

Going from G2.S4 to G0.S0 to is depicted by the sequence shown in Figure 33-1, "ACPI Cold Boot Sequence," beginning in the G2.S5 state. There is no way to discern the G1.S4 state from the G2.S5 state from merely observing the SoC signals.

## 33.9 Power Failure: From Gx.Sx To G3

While the SoC is capable of withstanding uncontrolled shutdowns due to a power failure, designs should adhere to the specified controlled power shutdown sequencing in order to avoid potential shortened component life expectancies. Intel does not specify or validate the number of power failure cycles a component may withstand before expected component life is affected.

In order to ensure that a power failure does not damage the SoC, the RSMRST_N signal must be asserted (must go low - from a 1 to a 0) at least 1µs before the main power supply that is used to derive all the SoC voltage rails falls 5% below its nominal value. With this sequence, initial testing has shown that components which have experienced 2000 uncontrolled shutdowns (power failures) have not experienced any failures or suffered shortened life expectancies.

## 33.10 Special Considerations for Platform Voltages

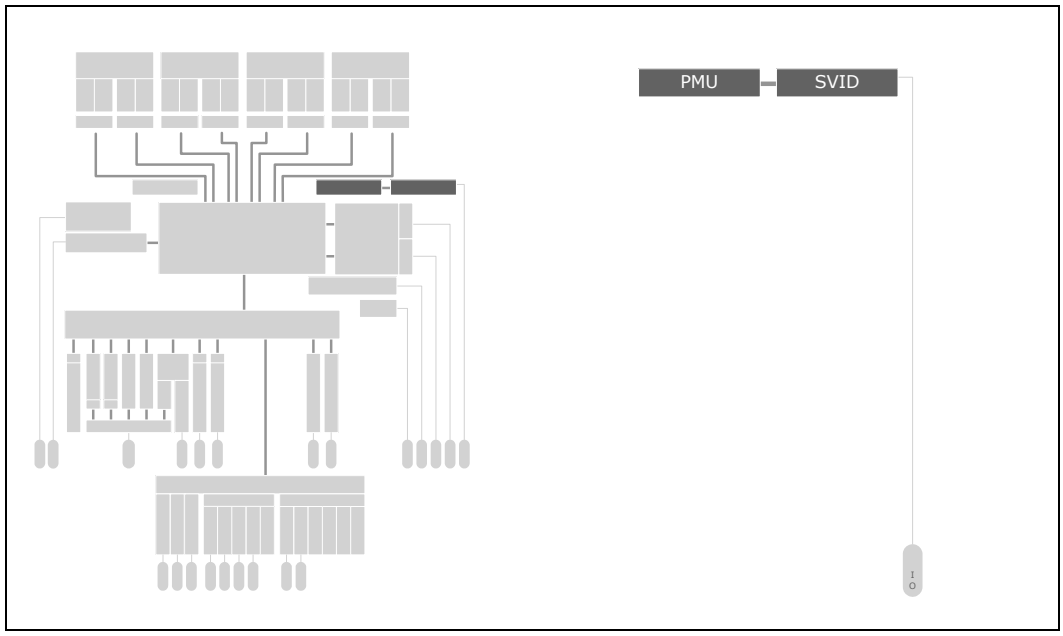
Voltages that do not directly connect to the SoC, but power other platform devices which connect to the SoC, must be isolated (to prevent leakage into SoC), until the SoC has powered up. The 5VDC supply which is provided to a USB interface is an example of a voltage that is not required by the SoC.

§ §

# 34 Power Management

The SoC provides dynamic power management that fits a number of usages like Microserver, Cloud Storage and Communication Infrastructure. As the technology changes and the number of cores per node and per module change, a better approach is required to manage optimal power of the node/module with respect to the power budget.

**Figure 34-1. What is Covered in This Chapter**




## 34.1 Signal Descriptions

The signal descriptions are shown in Table 34-1, "sVID Controller Signals." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a customer GPIO signal, see Chapter 39, "Signal Pin Names and Signal MUXing." The Direction/Type column of Table 34-1, "sVID Controller Signals" is interpreted as follows:

- I = Input signal. The SoC has receiver circuitry.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 34-1. sVID Controller Signals**

| Signal Name | Direction/ Type | Shared | Description |
|---|---|---|---|
| SVID_DATA | I,O-OD | Yes | **sVID Data:** Used by the SoC to send requests and data to the external Voltage Regulator (VR) and then by the VR to respond. |
| SVID_CLK | O-OD | Yes | **sVID Clock:** sVID requests driven on SVID_DATA by the SoC use this clock. The VR uses this clock to register-capture the requests. When the VR responds with data on SVID_DATA, the VR uses this SoC-driven clock. |
| SVID_ALERT_N | I | Yes | **sVID Alert:** Used by the VR to signal to the SoC that the prior request has not reached the requested operating point. |

## 34.2 Overview

The power management control is comprised of a number of Intel proprietary mechanisms. The power management signal interface to the rest of the platform is through the Power Management Controller (PMC) block. This chapter refers to this distributed function as the SoC Power Management Unit (PMU).

To take advantage of all power management capabilities an external, board-level Baseboard Management Controller (BMC) or some other embedded controller, is required to manage the platform power planes, power-on, sleep states, and reset signaling. This chapter refers to the BMC or Embedded Controller (EC). The EC interfaces with the SoC internal fabric to perform its management functions.

The SoC power management is responsible for the following tasks:

- Communicating with the EC and the Intel® Management Engine (Intel® ME)
- Reset sequencing
- Managing processor C-states and P-states
- Managing Sleep-state entry sequences
- Managing DDR4 power management
- Controlling Low Pin Count (LPC) interface clock
- Directing processor thermal management and memory
- Interfacing with the BIOS and the operating system software
- Managing Intel® Turbo Boost Technology, SoC RAPL control, and DRAM RAPL control

**Table 34-2. References**

| Description | Document ID / Location |
|---|---|
| *Advanced Configuration and Power Interface Specification Version 5.0, Errata A* | http://uefi.org/specifications |

## 34.3 Power Management Features

The power management features are:

- ACPI system power states supported: G0 (S0), G1 (S4), G2 (S5), G3 (Mechanical Off)
  - S1, S2, S3 are not supported.
- ACPI processor (CPU) C-states: C0, C1, C2, and C3.
- ACPI device states: D0, D3
- ADR (Asynchronous DRAM Refresh)
- Enhanced Intel SpeedStep® Technology
- Clock and Power gating
- Thermal throttling
- Dynamic I/O, memory, and SoC power reductions
- Running Average Power Limiter (RAPL)
- DDR4 SDRAM memory controller and PHY:
  - CKE active power-down (APD) and pre-charge power-down (PPD).
  - Self-Refresh
  - DDR PHY optimization to reduce SoC power.
  - DLL master/slave shut down based on CPU state.

### 34.3.1 Asynchronous DRAM Refresh (ADR)

ADR provides an early warning to the SoC of an impending reset or power failure so that it can save critical RAID information to the system memory DIMMs that remain powered by a dedicated battery while the reset is in progress.

See Section 16.15, “Asynchronous DRAM Refresh (ADR)” in this document.

## 34.4 Power Wells

The SoC circuitry is powered by three power wells:

- Core power well
- SUS power well
- RTC power well

### 34.4.1 Core Power Well

This power group includes all voltage rails and associated power wells that are on only when the system is in the S0 state (system is fully powered-on). The core power well contains the MAIN and cache power rails. These voltage rails are turned off when the system transitions to one of the other low-power system sleep states. See Figure 34-2, "Global System Power States and Transitions" for the ACPI states flow diagram.

### 34.4.2 SUS Power Well

The register-description chapters of the EDS use the terminology "Primary" and "DSW" for this power well instead of SUS Power Well.

This power group includes voltage rails and associated power wells that are on when the system is in the S0 and S5 states. These voltage rails are turned off when the system transitions from S5 to the G3 (mechanically off) state.

Most of the power management signal pins have their drivers/receivers in the SUS power well.

### 34.4.3 RTC Power Well

This power group includes all internal voltage rails and associated power wells that are always on, even when the system is in the G3 (mechanical off) state. This group is supplied its 3.3V from the SUS power supply, when the system is in S0 and S5. When the system is in the G3 state, the optional RTC power well is powered from an external battery source, typically a 3.0V lithium-type coin cell.

If a complete power failure (no AC power and no battery back-up supply) occurs, this voltage rail does not provide any power if there is no functioning battery providing its power. In the G3 state (mechanical off), it is permissible for designs not to have an external coin-cell battery if the design does not need to preserve information when the system power is turned off.

## 34.5 Supply Voltage Rails

The SoC is a highly integrated component where many traditional subsystems are contained on one die.

A total of nine external unique rails need to be supplied. Table 34-3, "SoC Voltage Rails" shows each voltage rail level and a list of internal units associated with the voltage rail.

See Table 33-3, "SoC Required Power Supplies" for voltage values for VCC, VNN,VCCRAM, and VCCREF.

**Table 34-3. SoC Voltage Rails**

| Voltage Rail | VR Rail | SoC Internal Units | S0 | S5 | RTC |
|---|---|---|---|---|---|
| Coin Cell Battery | VRTC | Real Time Clock (RTC) | On | On | On |
| sVID VCC VR | VCC | CPU Core | On | - | - |
| sVID VNN VR | VNN | UnCore | On | On | - |
| P1V05 | P1V05 | • GPIOSUS signal pins<br>• GbE control and I/O<br>• Power Management Controller (PMC) | On | On | - |
| VCCRAM | VCCSRAM | • Core L2 Cache | On | - | - |
| P1V8 | V1P8 | • GPIO SUS<br>• USB 2.0 SUS<br>• GbE Biasing | On | On | - |
| VDDQ | VDDQ | DDR I/O | On | - | - |
| VCCREF | | | On | - | |
| V3P3 | V3P3 | • GPIO SUS-Well High-Voltage signal pins<br>• USB 2.0 SUS signal pins<br>• Thermal Sensor<br>• USB2 I/O | On | On | - |

*Note:* S3 and S4 are part of the state machine and are included in Table 33-4, "SoC Power Supply for each Supported Power State." These states, however, are passed through on the way to S0 or S5.

RTC - In the G3 (mechanically off) state remains on when the RTC battery is present.

## 34.6 Serial Voltage Identification (sVID) Controller

The three signals provided by the SoC are in Table 34-1, "sVID Controller Signals."

### 34.6.1 SVID VR Requirements

It is required that the voltage rails for the cores (VCC), L2 SRAM (VCCRAM), and fabric (VNN) support IMON as defined by the Serial VID (SVID) Protocol Specification, Revision 1.7 which covers VR13.0.

The VDDQ VR, if used, must support IMON for DRAM RAPL and DRAM Energy Reporting to function. It is not, however, required for silicon functionality.

Failure to support IMON for the core voltage rails and VDDQ (if the VR is used) may result in one or more of the following:

- System fails to boot
- Power management features fail to work (e.g. RAPL)
- SoC reliability may be compromised
- SoC may show instability at varied temperatures

*Note:* Voltage control, VCC, VNN, and VCCSRAM, using the SVID feature is required for the silicon to function. Voltage control for VDDQ is supported but not required.

#### 34.6.1.1 sVID Commands

The sVID commands are shown in Table 34-4, "sVID Commands."

**Table 34-4. sVID Commands**

| Command | Command Code | Send Payload | Receive Payload | Description |
|---|---|---|---|---|
| Not Supported Reserved | 00h | (Extended Command) | | |
| SetVID_Fast | 01h | VID | n/a | Sets the new VID target. VR Jumps to the new VID target with controlled (up or down) slew rate programmed by the VR. When the VR receives a VID Moving Up command it exits all low-power states to the normal state to ensure the fastest slew to the new voltage. VR sets VR_settled bit and issues alert when VR has reached new VID target. |
| SetVID_Slow | 02h | VID | n/a | Sets the VID target. VR Jumps to new VID target with controlled slew rate (up or down) programmed by the VR. SetVID-Slow is 4x slower than SetVID-fast. When VR receives a VID Moving Up command it exits all low-power states to the normal state to ensure the fastest slew to the new voltage. VR sets VR_settled and issues alert when VR has reached new VID target. |
| SetVID_Decay | 03h | VID | n/a | Sets the VID target. VR jumps to new VID target, but does not control the slew rate, the output voltage decays at a rate proportional to the load current. SetVID_Decay is only used in VID down direction. VR sets VR_settled bit is set, but alert line is not asserted for SetVID-decay. |
| SetRegADR | 05h | VR Register Address | n/a | Sets the address pointer in the data register table. Typically the Next command SetRegDAT is the payload that gets loaded into this address. However, for multiple writes to the same address, only one SetRegADR is needed. |
| SetRegDAT | 06h | VR Register Data | n/a | Writes the contents to the data register that was previously identified by the address pointer with SetRegADR. |
| GetReg | 07h | VR Register Address | VR Register Data | Slave returns the specified register contents as the payload. The majority of the VR monitoring data is accessed through the GetReg command. |
| Test_Mode | 08h | | | Vendor Defined |
| Reserved | 08h through 1Fh | | | |

## 34.7 Core Power Management Overview

The SoC supports the hardware C0, C1, C1e, and C6 C-states. C3 is not supported. For C-states deeper than C6, the SoC will be automatically demoted to C6.

All cores in the C0, C1, or C1e C-state operate at the same voltage and frequency. The target frequency is typically the maximum requested frequency for all threads.

The SoC may or may not provide the requested frequency. Many factors can constrain the resolved frequency including but not limited to the following.

- Temperature
- Power constraints
- SKU constraints

The SoC determines a “resolved” frequency based on the OS requests and these constraints. The SoC autonomously manages the voltage based on the resolved frequency and various additional constraints.

The upper and lower bounds of these targets are weighed and prioritized to come up with a final ratio resolution for each core and the sections.

## 34.8 System Global Power States

Figure 34-2, "Global System Power States and Transitions" shows the system Global Power States as defined by the *Advanced Configuration and Power Interface Specification Version 5.0, Errata A*.

Table 34-5, "ACPI Power States" provides an overview of the power states defined for platforms based on the SoC. For design details on the power states definition and initiation see Chapter 33, "Resets and Voltage Sequencing."

**Figure 34-2. Global System Power States and Transitions**

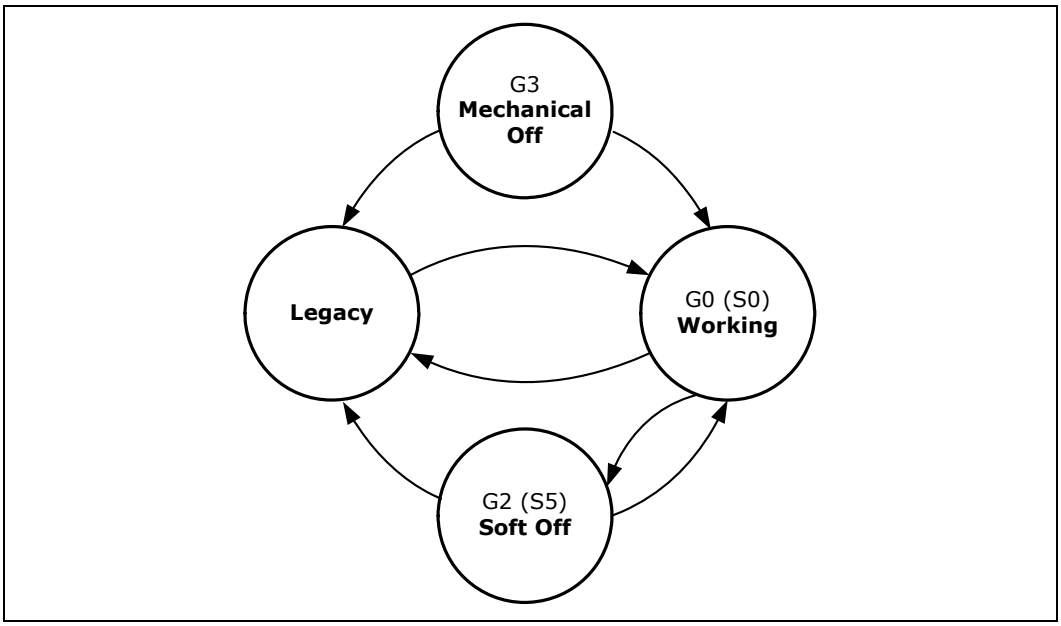



**Table 34-5. ACPI Power States**

| State (Global/Sleep/CPU) | APCI Name | Description |
|---|---|---|
| G0/S0/C0 | Working (Full On) | CPU operating. Individual devices are shut-off to save power. |
| G0/S0/C1<br>G0/S0 | Working<br>with CPU Power Management | The different CPU operating levels are defined by Cx states. |
| G2/S5 | Soft Off | System context not maintained. All main power is shut off except for the logic required to wake system. A full boot is required when waking. |
| G3 | Mechanical Off | • System context not maintained.<br>• All main power shut off except for the RTC.<br>• No Wake events occur, because the system does not have any power.<br>This state occurs if user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the waking logic.<br>When system power returns, transition depends on the state before entry to G3. |

**Table 34-6. ACPI Power State Transitions for the SoC**

| State (Global/Sleep/CPU) | Transition Trigger | Next State |
|---|---|---|
| G0/S0/C0 | Executing the MWAIT instruction or LVL Rd | G0/S0/Cx |
| | Sleep Enable (SLP_EN) bit written to 1 by the software in the Power Management 1 Control (PM1_CNT) register | Specified by the 3-bit code Sleep Type (SLP_TYP) field of the PM1_CNT register:<br>• 000 G0/S0 - Working (Full On)<br>• 001 G1/S1 - Not supported by the SoC<br>• 010 Reserved<br>• 011 Reserved<br>• 100 Reserved<br>• 101 G1/S3 - Suspend-To-RAM - Not supported by the SoC<br>• 10 G1/S4 - Suspend-To-Disk S4<br>• 111 G2/S5 - Soft Off |
| | Power Button Override | G2/S5 |
| | Mechanical Off/Power Failure | G3 |
| G0/S0/C1<br>G0/S0/C3 | C-State break events including:<br>• CPU Snoop<br>• MSI<br>• Legacy Interrupt<br>• Always-On (AONT) Timer expires | G0/S0/C0 |
| | Power Button Override | G2/S5 |
| | SUS Well Power Failure | G3 |
| G2/S5 | Any Enabled Wake Event | G0/S0/C0 |
| | SUS Well Power Failure | G3 |
| G3 | Power Returns | Option to go to:<br>• S0/C0 (Reboot)<br>• G2/S5 (Stays off until the power button is pressed or other enabled wake event) |
| S0/G3/S0 | Surprise Power Loss | ADR (Automatic DIMM Refresh) asserts if the appropriate battery back-up design is in place. |

## 34.8.1 Idle Power Optimizations

The SoC does not use the G1/S1 sleep state. Instead, as other recent server processors do, it enters the G1/S1 sleep state is not used. The processor now enters low power states (package C-states) while staying in the S0 state. This method provides low latency wake-ups. Alternatively, the S4 (sleep) or S5 (soft off) may be used to put the system into its lowest idle power states. Note the exit latency between S4 and S5 are OS and platform configurations dependent. In certain cases, S4 latency is not necessarily faster than S5.

## 34.9 Processor Power States - C-States

There are three types of C-states which are frequently mixed up:

- ACPI C-states
- Core C-states
- Package C-states

ACPI C-states are C0, C1, C2, and C3. These states are mapped to the actual hardware C-states (C0, C1, C1e, C6) through BIOS tables.

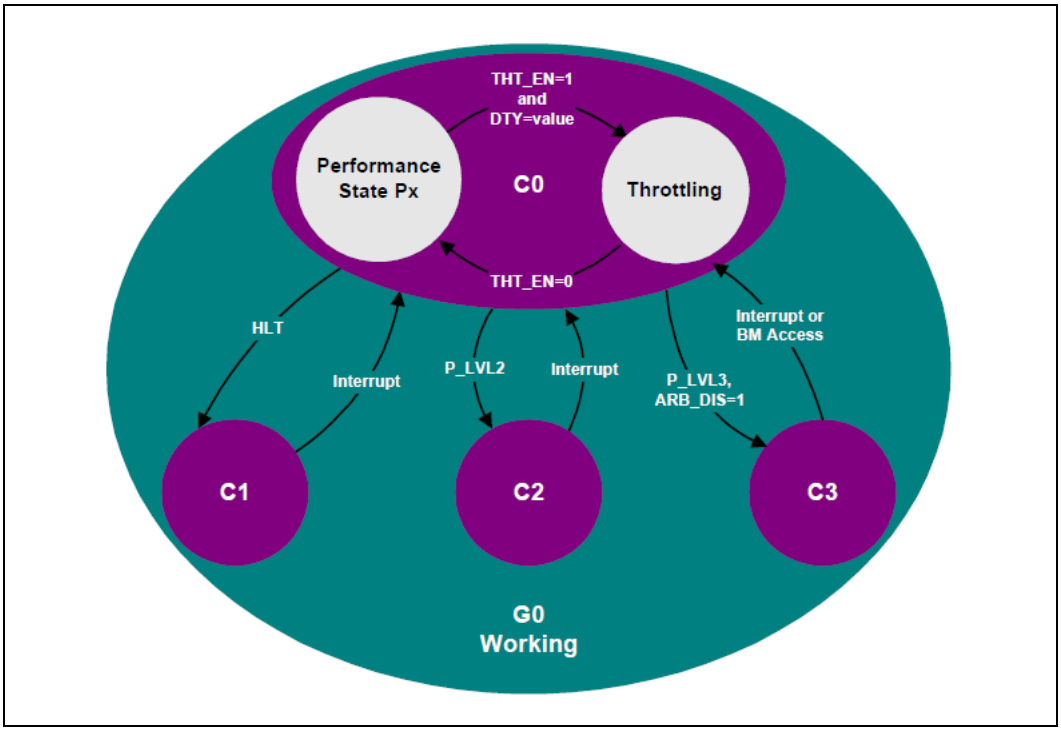
When all cores go into certain Core C-states, the SoC has the option to enter a deeper package C-state and take additional power saving actions in the package and/or on the platform.

*Note:* Linux and the mainline kernel have moved away from ACPI C-states with the intel_idle driver. This driver will be updated to support this SoC at launch.

Figure 34-3 shows the Processor Power States described in the *Advanced Configuration and Power Interface Specification Version 5.0, Errata A*.

The processor Performance States (P-States) are described later in Section 34.10.1, "Processor Performance States - P-States."

**Figure 34-3. Processor Power States**



Concerning the SoC CMP Module and its interface to the System Agent, the SoC has three types of C-States:

- Core C-States. See Table 34-7, "Core C-States."
- Package C-States. See Table 34-8, "Package C-States."

### Table 34-7. Core C-States

| HW Core C-State | Typical Mapped ACPI C-State | Description |
|---|---|---|
| CC0 | C0 \| ACPI C0 | Active State |
| CC1 | C1 \| ACPI C1 | Some core clocks gated.<br>L1 Data Cache Snoops are serviced. |
| CC1e | C1e \| N/A | Some core clocks gated.<br>L1 Data Cache Snoops are serviced.<br>Provides a hint to reduce voltage and frequency. |
| CC6 | C6\| ACPI C2 | Core power gated and clock gated L1 Data cache is flushed |

### Table 34-8. Package C-States

| Core C-State | Package C-State | When Entered |
|---|---|---|
| C0 | PC0 | • Active States<br>• When at least one core is in CC0 |
| C1e | PC1e | • Reduce voltage and frequency for C1e<br>• When no cores are in CC0 or CC1 and at least 1 core is in C1e |
| CC6 | PC2 | • All cores in CC6<br>• Some package actions allowed |

## 34.10 Performance States

This section describes the concept of processor and device performance states. Processor and device performance states (Px states) are power consumption and capability states within the active/executing states, C0 for processors and D0 for devices. Performance states allow the Operating System Power Management (OSPM) to make trade-offs between performance and energy conservation. Processor and device performance states have the greatest impact when the states invoke different device and processor efficiency levels as opposed to a linear scaling of performance and energy consumption. Since performance state transitions occur in the active/executing device states, ensure that performance state transitions do not adversely impact the system.

### 34.10.1 Processor Performance States - P-States

The SoC supports P-States for every dual-core pair within a module. Based on power performance analysis, the SoC only supports Package Level P-States. The Power Management Unit will select the highest P-state from all requests across all modules and apply that state to all cores.

*Note:* Hardware Power Management and Hardware P-states are not supported.

#### 34.10.1.1 Frequency/Voltage Scaling

P-state frequency and voltage are resolved at the package level and not by the individual cores or modules.

**Table 34-9. ACPI P-State Mappings**

| P-State | ACPI Meaning | Frequency Mapping |
|---|---|---|
| P0 | Performance is preferred over power efficiency | Greater-than or equal-to the Maximum Non-Turbo Limit Ratio.<br>This is set based on thermal/electrical limits, platform constraints, and other parameters. |
| P1 | Maximum performance/efficiency is desired | Maximum Non-Turbo Limit Ratio (base frequency) |
| P2 through PN | Intermediate performance/efficiency is desired | Less-than the Maximum Non-Turbo Limit Ratio,<br>Greater-than the Maximum Efficiency Ratio |
| PN | Frequency at the minimum voltage | Maximum Efficiency Ratio correlates to minimum operational voltage (LFM_RATIO) |

The OS requests a P-State based on application performance needs. A desired P-State is requested via IA32_PERF_CTL (CLOCK_CR_GEYSIII_CONTROL). The SoC supports the Enhanced Intel SpeedStep® Technology.

### 34.10.2 Software P-State Requests

ACPI enumerates the base frequency at P1 and the maximum turbo frequency at P0. There are multiple ratios/frequencies between these two points. The SoC supports requests to intermediate ratios and will attempt to provide those ratios (under all normal constraints).

## 34.11 Power Management Technologies

### 34.11.1 Intel® Turb Boost Technology

Intel® Turbo Boost Technology enables higher performance through the availability of increased core frequency under certain configurations and workloads. Turbo allows processor cores to run faster than the specified operating frequency if the processor is operating below rated power, temperature, and current specification limits of the system. Turbo is engaged with any number of cores or logical processors enabled and active, enabling increased performance of both multi- and single-threaded workloads.

The BIOS may disable turbo features on turbo-enabled SKUs. If Reset = 1, then the SKU does not support turbo and it is disabled. If RESET = 0, then the SKU does support turbo and it is enabled by default. BIOS can set the bit to disable it.

The Operating System (OS) and applications must use CPUID.06H:EAX[1] to detect whether the BIOS has enabled turbo features. If IA32_MISC_ENABLES[38] is set, CPUID.06H:EAX[1] returns 0. Turbo features have been enabled by BIOS.

Because IA32_MISC_ENABLES[38] is defined per-package, CPUID has to read from the UnCore to get MISC_ENABLES[38]. The OS or BIOS manages read-modify-write conflicts cross-core. Setting IA32_MISC_ENABLES[38] on any core causes the SoC to disable turbo operation for ALL cores.

Certain software workloads may not be able to tolerate the non-deterministic aspects of turbo operation. The software temporarily disables turbo operation by setting CLOCK_CR_GEYSIII_CONTROL[32] bit of the IA32_PERF_CTL Model-Specific Register (MSR) [MSR 0199h]. As previously mentioned, disabling turbo on any core causes turbo to be disabled for ALL cores.

Two MSRs (MSRs 0x1AE and 0x1AD) can be used to limit the maximum amount of turbo that is granted based on the number of active cores. At reset these registers are populated based on the SKU limits. Software (either BIS or Ring-O) can reprogram these limits at boot or runtime. Configuring frequencies larger than the SKU limits will be silently clipped.

### 34.11.2 Voltage Regulator Constraints

For some platforms, the voltage regulator for the shared VCC voltage rail has some maximum current limits. A typical voltage regulator for a platform using the SoC has a Thermal Design Current (Icctdc) specification and a Maximum Current (Iccmax) limit.

Icctdc represents the current which is sustained indefinitely by the voltage regulator (i.e., able to sustain a TDP workload on all processors running at base frequency), whereas the Iccmax is the peak current which the voltage regulator does source without tripping any protective circuitry. Any current greater than Icctdc is only sustained for a short duration. Iccmax is sustained typically for approximately 10 ms.

Such voltage-regulator specifications must be taken into account when configuring the aggressiveness of turbo operation for the device. Turbo operating points are limited based on the number of active cores to keep the current drawn within the Iccmax limit.

See Section 35.2, "Feature Summary" for Thermal Management Features.

### 34.11.3 Thermal Design Power Constraints

The Thermal Design Power (TDP) represents the power consumed by the device when running a realistic worst-case workload at $T_{J\text{-}MAX}$ temperature. The realistic-worst-case workload is determined based on knowledge of the target application and realistic usage scenarios. The system thermal solution must be designed to dissipate the heat generated during a sustained period of activity with all cores running at TDP at guaranteed maximum frequency.

See Section 35.2, "Feature Summary" for Thermal Management Features.

### 34.11.4 FAST PROCHOT

Fast PROCHOT provides a mechanism for the platform to trigger a very quick and heavy throttling (decreasing the clock frequency) of the SoC processor cores. This description is an overview of the functionality. For design details for PROCHOT and Fast PROCHOT see Section 35.4, "Thermal Management (TM)."

Fast PROCHOT is typically used by the power supply unit (PSU) to trigger an immediate throttle to protect against unexpected current spikes or thermal conditions. Other customer platform innovation is possible.

Benefits:

- Enables PSU cost reduction without excessive guard banding lowering cost.
- Enables Processor performance without excessive guard banding enhancing performance.

High-Level Flow:

- Via a comparator at the PSU PROCHOT_N is asserted.
- The SoC Power Management Unit (PMU) utilizes the new fast throttle mechanism. Power is reduced quickly via core instruction throttling.

A frequency transition occurs later and is used to provide more power/performance efficient throttling.

### 34.11.5 Running Average Power Limiting (RAPL)

The SoC contains proprietary power monitors and Running Average Power Limit (RAPL) algorithms that calculate an energy budget and convert the budget into voltage/frequency working points. This section provides an overview of RAPL. For details see Section 35.6, "Run Average Power Limit (RAPL)."

The SoC supports RAPL control through these interfaces:

- MMIO. Recommended usage is BIOS and platform limits.
- MSR. Recommended usage is runtime limits through ring-0 software).
- PECI. Recommended usage is through an external Baseboard Management Controller (BMC) or other platform circuitry via the SoC PECI/SMBus interface.

RAPL is used in mobile devices to conserve battery power, but the SoC is not typically used in applications where battery is the main source of power. The SoC uses RAPL to deal with the SoC Package and DRAM thermal management in the high-density platform environment of Microserver, Storage, and Communication applications. See Section 35.2, "Feature Summary" for Thermal Management Features.

### 34.11.6 Always-On Timers (AONT)

Always-On Timers run while the SoC is in the S0 state and are used to periodically wake-up the cores from sleeping. They do not run when the SoC transitions out of S0.

### 34.11.7 I/O Device Controller Enable/Disable

**Table 34-10. I/O Power Management Summary**

| Integrated I/O Device | I/O Feature Not Used by Customer | Nothing Connected to Interface at Boot Time |
|---|---|---|
| PCIe* Root Ports | The BIOS disables the circuitry and power on a per-lane basis. | The BIOS disables the circuitry and power on a per-lane basis. |
| SATA 3.1 | The BIOS disables the circuitry and power-on a per-port basis. | Disabled at power-on unless explicitly enabled by the software. |
| LAN | The BIOS disables the circuitry and power-on a per-port basis. | Ports not WOL-enabled are in P2.<br>Ports with WOL are in P0.<br>If all ports are WOL-disabled, internal clocks off. |
| USB | The BIOS disables the circuitry and power. | Idle State with internal clocks off. |

§ §

# 35 Thermal Management

The SoC implements configurable forms of thermal management for itself, memory, and the system. The architecture implements various proven methods of maintaining maximum performance while remaining within thermal specifications. Thermal Control Circuit (TCC) mechanisms are used to reduce power consumption when thermal device limits are exceeded, and the system is notified of this condition via interrupts or thermal signaling pins.

## 35.1 Signal Descriptions

The signal descriptions are shown in Table 35-1, "Signal Names and Descriptions". For additional signal information including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing." The Direction/Type column of Table 35-1, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 35-1. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| MEMHOT_N | I | Yes | **Memory Hot:** Active-low input signal from the platform board indicating memory exceeds temperature limits. When this signal is active, the SoC can perform memory throttling in an attempt to cool the memory devices. |
| PROCHOT_N | I,O-OD | Yes | **Processor Hot:** Active low.<br>As an input, the platform board External Circuitry (EC) chooses to drive this signal low to reduce SoC current consumption if the EC detects overheating in the board or system.<br>Also as an input, the power supply asserts this signal to trigger the SoC Fast PROCHOT feature in the event of a maximum power excursion.<br><br>As an output, the signal is asserted if any SoC thermal sensor indicates the component is operating above temperature limits. The platform board must ignore this SoC output signal while PMU_PLTRST_N (active-low SoC output) is asserted.<br>***Note:*** PROCHOT_N can be configured to be an output or an input signal pin. It cannot be configured to be bi-directional. |
| THERMTRIP_N | O-OD | Yes | **Catastrophic Thermal Trip:** Active-low, open-drain output signal indicating the SoC has reached an operating temperature that may damage the component. Once THERMTRIP_N is asserted, the platform must remove power by shutting down the power regulators. The platform board must ignore this SoC output signal while PMU_PLTRST_N (active-low SoC output) is asserted. See Section 35.4.5, "THERMTRIP_N" for additional details on THERMTRIP_N.<br>Even though this open-drain output has an internal 20-kΩ pull-up resistor, an external 51-Ω pull-up resistor tied to V1P05 is recommended for open-drain operation. |

## 35.2 Feature Summary

Thermal management features include:

- Up to 13 Digital Thermal Sensors (DTS):
  - Eight sensors, one for each L2 Cache in the processor core area.
  - Five sensors not in the processor core area.
- Thermal Control Circuit (TCC) mechanisms:
  - Configurable PROCHOT_N.
  - Intel® Thermal Monitor 1.
  - Intel® Thermal Monitor 2.
- Catastrophic temperature protection (THERMTRIP_N) and memory temperature protection (MEM_THERMTRIP).
- Two programmable thermal interrupt thresholds per DTS.
- Memory Temperature Management control modes:
  - OLTT - Open Loop Thermal Throttling.
  - CLTT - Closed Loop Thermal Throttling via SMBus-Legacy interface.
  - CLTT Pass-through via SMBus-PECI interface.
- Thermal status/trigger signal pins:
  - MEMHOT_N input-only signal.
  - PROCHOT_N configurable as an input-only signal or as an output-only signal.
  - THERMTRIP_N output-only signal.
- Fan Speed Control (FSC) parameter ($T_{CONTROL}$ and margin to $T_{CONTROL}$)
- Fast PROCHOT_N input signal pin driven by the platform or power supply that triggers a rapid Intel® Thermal Monitor 2 (TM2) throttling response:
  - Available only when PROCHOT_N pin is configured as input-only signal.
- Inverse Temperature Dependence (ITD) function for maintaining processor performance at low temperatures.
- Platform Voltage Regulator Hot Alert to lower voltage needs when a VR is approaching its thermal limits.

## 35.3 Registers with Multiple Access Methods

Many of the Thermal Management registers mentioned in this chapter are accessible in more than one way. Even so, the type of access used may have unique Read/Write access capabilities. In some cases of access via the PECI SMBus, the temperature values may have a different format or bit length. For details see Chapter 30, "SMBus Controller - Platform Environment Control Interface."

## 35.4 Thermal Management (TM)

The IA-32 architecture provides mechanisms for monitoring temperature and controlling thermal power. These are presented in sub-section 14.7 of Volume 3 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual* and are listed below along with the section in this document:

- Catastrophic Shutdown detector
  See Section 35.4.5, "THERMTRIP_N."
- Automatic and Adaptive Thermal Monitoring mechanisms
  See Section 35.4.4, "Thermal Throttling."
- Software-Controlled Clock Modulation mechanism
  See Section 35.4.4.4, "Software-Controlled Clock Modulation."
- On-die Digital Thermal Sensor (DTS) and interrupt mechanisms
  See Section 35.4.1, "Digital Thermal Sensor (DTS)."

### 35.4.1 Digital Thermal Sensor (DTS)

Each of the 13 on-die Digital Thermal Sensor (DTS) reports the temperature of the associated processor core pair or other portions of the SoC circuitry as a temperature relative to the factory-configured Thermal Control Circuit (TCC) Activation Temperature which is also known as $DTS_{MAX}$.

- There is one DTS for each two-core processor. Here two cores share an L2 cache and a Bus Interface Unit (BIU). Some SoC product SKUs provide only one core processor per L2 cache and BIU and contain one DTS per core processor.
- There are five additional DTSs located in the remaining SoC circuitry.

Each core-related DTS produces a 7-bit temperature value in the TEMPERATURE field of its IA32_THERM_STATUS register (MSR 19Ch). The TEMPERATURE value is reported as a negative offset from the eight-bit REF_TEMP field of the TEMPERATURE_TARGET register. REF_TEMP is the Thermal Monitor Activation Temperature, also called the PROCHOT_N Temperature, TCC Activation Temperature, and $DTS_{MAX}$. The REF_TEMP value is the maximum temperature at which the silicon is capable of operating.

Example: If TEMPERATURE_TARGET.REF_TEMP indicates 90 °C, a IA32_THERM_STATUS.TEMPERATURE value of 10 decimal implies that this domain is currently operating at 80 °C.

The SoC Package temperature is based on all 13 DTSs and is in the TEMPERATURE field of the IA32_PACKAGE_THERM_STATUS register (MSR 1B1Ch). The TEMPERATURE value is reported as a negative offset from the aforementioned REF_TEMP field of the TEMPERATURE_TARGET register.

The registers associated with DTS configuration and target values are shown here. The core-related and SoC Package Status registers are mentioned in Section 35.5.3, "Thermal Status and Interrupts":

**TEMPERATURE_TARGET**

MSR_TEMPERATURE_TARGET (MSR 1A2h)
P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR (MMIO MCHBAR + 7074h)
Temperature Target Read Data (PECI RdPkgConfig(), Index 16)

This 32-bit register contains information about the Fan Speed Control (FSC) target temperature as well as details on the reference temperature for core-pair DTS relative temperature reading. The register contains the TJ_MAX_TCC_OFFSET. It also contains the Read-Only REF_TEMP (also known as $DTS_{MAX}$) and FAN_TEMP_TARGET_OFFSET (also known as $T_{CONTROL}$) fields which are set at the Intel factory.

DTS_CONFIG3_CFG

P_CR_DTS_CONFIG3_CFG_0_0_0_MCHBAR (MMIO MCHBAR + 7118h)

This 32-bit register contains an offset value to modify $T_{CONTROL}$. This register contains the legacy TCONTROL_OFFSET and OFFSET_PROGRAMMED fields.

## 35.4.2 Fan Speed Control (FSC)

*Note:* This chapter deals with software-controlled and BMC-controlled platform fan speed control. While the SoC has the FAN_PWM[3:0] and FAN_TACH[3:0] signal pins, the pins are exclusively controlled by the customer-programmed Innovation Engine (IE) and are not discussed here. See Chapter 37, "Innovation Engine."

While the SoC does not support the DTS 2.0 Fan Speed Control (FSC) feature, it does support the FSC as used on many Intel® Xeon® processor products that are multi-chip packages (MCP) even though the SoC is not an MCP.

The MCP FSC feature reports "Thermal Margin" as the temperature margin relative to the TEMPERATURE_TARGET.FAN_TEMP_TARGET_OFFSET. The FAN_TEMP_TARGET_OFFSET value is also known as $T_{CONTROL}$. It uses the same Thermal Margin access methods as DTS 2.0 FSC, but the underlying calculations are different and the overall algorithm is much simpler.

The Thermal Margin value simply reflects whether or not the SoC is meeting the TEMPERATURE_TARGET.FAN_TEMP_TARGET_OFFSET ($T_{CONTROL}$) specification; negative margin means the platform needs to increase fan speed. At high power levels, Thermal Margin becomes negative because of the thermal load line, but

TEMPERATURE_TARGET.REF_TEMP ($DTS_{MAX}$) is set at the Intel factory so that the SoC does not throttle when running a TDP workload and meeting the SoC $T_{CASE_MAX}$ specification.

Thermal Margin is calculated as:

PACKAGE_THERM_STATUS.TEMPERATURE
*minus the*
TEMPERATURE_TARGET.FAN_TEMP_TARGET_OFFSET
*where FAN_TEMP_TARGET_OFFSET is a positive value*

Software can read the value of Thermal Margin by PECI command or by the THERM_MARGIN field of an MSR or an MMIO register:

**PACKAGE_THERM_MARGIN**

MSR_PLATFORM_BRV (MSR 1A1h)
P_CR_PACKAGE_THERM_MARGIN_0_0_0_MCHBAR (MMIO MCHBAR + 703Ch)
THERM_MARGIN Read Data (PECI RdPkgConfig(), Index 10)

Bits 15:0 (of the MSR and MMIO) is the THERM_MARGIN: This product does not support DTS 2.0 and margin to the thermal loadline. This register reports margin to the T_CONTROL temperature in 2's compliment format (S16.7.8). Negative margin implies that the temperature is operating above the (static) T_CONTROL temperature, and that thermal controls (fans, etc.) should be attempting to cool the silicon. Positive margin implies that the silicon is operating at a temperature below the T_CONTROL point and that a reduction in cooling may provide better power efficiency. Unlike Intel products that support DTS 2.0, sustained negative margin is expected and acceptable when the SoC is operating with high power consumption.

### 35.4.3 PROCHOT_N Signal Pin

The SoC PROCHOT_N signal pin can be configured to be an output or an input signal pin. It cannot be configured to be bi-directional. Bits 0, 21, 22 and 23 of the MSR_POWER_CTL register, MSR 1FCh, provide this configuration.

See Table 35-2, "MSR_POWER_CTL (MSR 1FCh)."

**Table 35-2. MSR_POWER_CTL (MSR 1FCh)**

| Bits | Access Type | Default | Description |
|---|---|---|---|
| 31:25 | - | - | *Reserved* |
| 24 | RW | 0 | **VR_THERM_ALERT_DISABLE**: When set to 1, disables the VR_THERMAL_ALERT signaling. |
| 23 | RW | 0 | **PROCHOT_LOCK**: When set to 1, locks bits 0, 21, 22 and 23 of this MSR. Once set, a reset is required to clear this bit. |
| 22 | RW | 0 | **PROCHOT_RESPONSE**: PROCHOT_N signal pin configurable response enable.<br>0 = Go to Pn on incoming asserted PROCHOT_N signal.<br>1 = Go to a state lower than Pn on incoming asserted PROCHOT_N signal. Go to Pm on incoming PROCHOT_N.<br>***Note:*** PROCHOT_OUTPUT_MODE_DISABLE must be 1 (input-only signal pin) for this bit to function. |
| 21 | RW | 1 | **PROCHOT_OUTPUT_MODE_DISABLE**: PROCHOT_N signal pin output disable.<br>0 = PROCHOT_N is an output-only signal pin.<br>1 = PROCHOT_N output is disabled and the PROCHOT_N signal pin is input-only. |
| 20:19 | - | - | *Reserved* |
| 18 | RW | 0 | **PWR_PERF_PLTFRM_OVR**: When set to 0 (default), this bit disables access to IA32_ENERGY_PERF_BIAS (MSR 1B0h), causing a GP# fault on access. CPUID.(EAX=6):ECX[3] will read 0. |
| 17:2 | - | - | *Reserved* |
| 1 | RW | 1 | **C1E_ENABLE**: Used to enable C1E behavior on every C1 entry (even if MWAIT C1E sub-state bit was not set or when using HLT). When this package-visible bit is set, it causes the CPU to switch to minimum GV point when all cores have entered C1. |
| 0 | RO | 0 | *Reserved* |

The PROCHOT_N signal direction and how the SoC reacts are shown in Table 35-3, "PROCHOT_N Signal Direction and Function."

**Table 35-3. PROCHOT_N Signal Direction and Function**

| MSR_POWER_CTL Bit 21 PROCHOT_OUTPUT_MODE_DISABLE | MSR_POWER_CTL Bit 22 PROCHOT_RESPONSE | Resulting PROCHOT_N Signal Pin |
|---|---|---|
| 0 | x | Output-Only Signal |
| 1 | 0 | Input-Only Signal, go to Pn (the default mode) |
| 1 | 1 | Input-Only Signal, go to Pm |

- Pn - Use-frequency scaling and voltage scaling to attain PROCHOT relief. Pn is the lowest frequency of operation, also called the Low Frequency Mode (LFM) frequency.
- Pm - Provides more PROCHOT relief than Pn mode does.

The default mode for PROCHOT_N provides the Fast PROCHOT capability. Customers can use the Input-Only, Pm mode to get some additional power savings on PROCHOT, but this will be mostly frequency-only scaling and not voltage scaling.

Generally, Pm provides a small amount of additional PROCHOT relief versus Pn. Even so, the Pm mode can help a bit more on SoC product SKUs that have small dynamic range from throttling.

Fast PROCHOT is enabled when the MSR (1FCh) MSR_POWER_CTL field PROCHOT_OUTPUT_DISABLE (bit 21) is set. When Fast PROCHOT enabled the processor responds to platform requests to operate within the platform power budget, when rapidly exercised by non-ideal workloads. The PROCHOT_N input time response has been improved on the SoC from 3ms to 80us (from PROCHOT_N pin assertion), when FAST PROCHOT is enabled. Additional time is required for the platform to detect a Pmax condition and assert the PROCHOT_N pin (typically around 20 uS). If a rise in power at or above 1.6x TDP for the processor is detected by platform logic, a PROCHOT_N input by the platform or power supply triggers this rapid Intel® Thermal Monitor 2 (TM2) throttling response. Platform logic is required to detect the power level within 20µs and assert PROCHOT_N for this feature usage. It is recommended that PROCHOT_N be configured for input-only mode to take advantage of this reduced response time. The platform is required to use DTS to check for over temperature conditions, when Fast PROCHOT is enabled as the primary means of temperature monitoring for thermal control, as the PROCHOT_N signal output is disabled.

The Output-Only mode can be useful on systems that want to use PROCHOT_N for fan speed control. While this is not the recommended usage model, it can be used on very-low-cost platform designs. PROCHOT_N is asserted by the SoC when a thermal sensor indicates the component is operating above temperature limits.

## 35.4.4 Thermal Throttling

The SoC has only one thermal-control domain—the processor cores. No uncore throttling is supported. When an over-temperature event occurs based on the Throttle Threshold Temperature (see Section 35.4.4.1, "Throttle Threshold Temperature"), all processor cores are throttled regardless of which of the 13 DTS within the SoC is reporting the over-temperature event.

*Note:* The SoC does not throttle the processor when DRAM Memory is hot. Memory Thermal Management (see Section 35.5, "Memory Thermal Management (MTM)") is managed independently of Thermal Management.

Two Thermal Throttling mechanisms are implemented to reduce temperature by reducing power consumption in response to temperatures exceeding the Throttle Threshold Temperature:

- Intel® Thermal Monitor 2 (TM2) - Thermal throttling based on voltage and frequency reductions. See Section 35.4.4.2, "Core Frequency/Voltage Reduction (Intel® Thermal Monitor 2)."
- Intel® Thermal Monitor 1 (TM1) - Thermal throttling via core clock modulation using the processor's eHALT feature. See Section 35.4.4.3, "Clock Modulation (Intel® Thermal Monitor 1)."

*Note:* IA32_MISC_ENABLE (MSR 1A0h), bit 3, is the Automatic Thermal Control Circuit (TCC) Enable. The SoC ignores the setting of this MSR bit, even if it is 0 and the SoC always invokes TM2 and TM1 (automatic TCC).

The SoC invokes TM2 when the SoC package temperature (the highest temperature of the 13 DTSs) has met or has exceeded the Throttle Threshold Temperature defined in the next sub-section.

The SoC only invokes TM1 if TM2 has reached its Maximum Throttle Point of Pm and the SoC is still over temperature.

### 35.4.4.1 Throttle Threshold Temperature

The Throttle Threshold Temperature is determined from the following equation:

TEMPERATURE_TARGET.REF_TEMP — TEMPERATURE_TARGET.TJ_MAX_TCC_OFFSET

TJ_MAX_TCC_OFFSET can be set by software in 1°C increments. The register default value is 0 offset. If set to 0, the SoC uses an internally-set guardband offset of approximately 0.5°C, making the Throttle Threshold Temperature about 0.5°C below the REF_TEMP value (REF_TEMP is also known as the $DTS_{MAX}$ temperature or the PROCHOT temperature).

### 35.4.4.2 Core Frequency/Voltage Reduction (Intel® Thermal Monitor 2)

Intel® Thermal Monitor 2 (TM2) refers to thermal throttling based on voltage and frequency. The SoC attempts to lower the temperature by transitioning to a lower operating frequency and lower operating voltage. An internal algorithm attempts to select the highest possible frequency while staying within thermal constraints. If enabled, TM2 is automatically selected as the primary thermal control mechanism and cannot be disabled by the Operating System software.

Since the SoC does not support per-core Px states, all processor core modules of the SoC package throttle together based on the Throttle Threshold Temperature.

When TM2 starts it steps-down the frequency/voltage until the Maximum Efficiency Throttle Point (Pn) is achieved. If the SoC package temperature is still equal-to or above the Throttle Threshold Temperature, it continues to step-down to the Maximum Throttle Point of Pm. If the package temperature is still not below the Throttle Threshold Temperature, TM1 is invoked.

### 35.4.4.3 Clock Modulation (Intel® Thermal Monitor 1)

Intel® Thermal Monitor 1 (TM1) is activated as a secondary control mechanism when TM2 (see Section 35.4.4.2, "Core Frequency/Voltage Reduction (Intel® Thermal Monitor 2)") is unsuccessful at reducing processor temperatures, and the SoC is still over temperature. TM1 cannot be disabled by the Operating System software.

**Figure 35-1. TM1 Use of Processor Clock Duty Cycle Modulation**

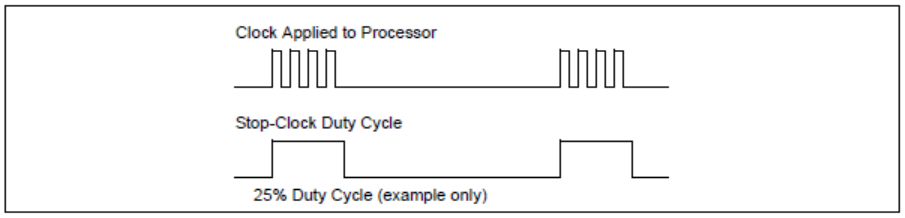



Figure 14-21. Processor Modulation Through Stop-Clock Mechanism

TM1 works by periodically stopping (gating) the processor clock for a period of time (microseconds), preventing all execution and reducing power. The "duty cycle" of TM1 refers to the percentage of time the clock is running and not gated off.

When TM1 throttling is invoked, the core clock duty-cycle modulation starts stepping down from the clock's maximum "on" duty-cycle ratio until the clock's minimum "on" duty-cycle ratio (1/8) is reached. During pre-throttle, it steps-down the "on" duty cycle 12.5% once every 8 ms. During throttle it does this more often: once every 1 ms.

When TM1 throttling is reduced during throttle the clock "on" duty-cycle ratio goes back to every 8 ms (pre-throttle). If throttling is further reduced, the "on duty-cycle ratio returns to the maximum.

The stop-clock time duration is clock-frequency dependent. Higher frequencies stop the clock for shorter time durations for a given "on" duty-cycle configuration.

#### 35.4.4.4 Software-Controlled Clock Modulation

This mechanism permits Operating Systems (OS) to implement power-management policies that reduce power consumption. This is in addition to the reduction offered by automatic thermal monitoring mechanisms TM2 and TM1.

The SoC supports the "Extension of the software controlled clock modulation" described in sub-section 14.7 of Volume 3 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

The mechanism is accessible to software through the IA32_CLOCK_MODULATION register (MSR 19Ah).

The register fields are:

— THROTTLE_ACTIVE (bit 4) enables the Software-Controlled Processor Clock Modulation mechanism.

— T_STATE_REQ (bits 3:0), when THROTTLE_ACTIVE is set, this 4-bit field selects the clock modulation duty cycle in 6.25% increments.

For multiple processor cores in a physical package, each processor core can modulate to a programmed duty cycle independently.

This mechanism functions much like the automatic hardware-controlled TM1 mechanism in reducing SoC power consumption.

### 35.4.5 THERMTRIP_N

The THERMTRIP_N feature provides SoC protection in the event of a catastrophic thermal event, i.e., temperatures are reached which may damage the SoC silicon. The temperature to trigger THERMTRIP_N is set at the factory and is higher than the temperature that invokes TM2 and TM1. The protection is accomplished by reducing SoC power as fast as possible. Dynamic power is largely eliminated by the SoC within microseconds by means of shutting down as many internal clocks as possible.

The SoC asynchronously signals the platform with PMU_SLP_S45_N indicating the SoC has entered the low power S5 state and logs an error in the SoC circuitry in the RTC Power Well.

*Note:* THERMTRIP is one of the Power Button Override Events. These events cause an unconditional transition to the S5 state and the Power Button Override (PWRBTNOR_STS) bit of the Power Management 1 Enables and Status (PM1_EN_STS) register is set by the SoC and saved by the SoC RTC power well. This bit is not affected by hard resets via CF9h I/O writes and is not reset by the internal POWERGOOD reset. The THERMTRIP indication is preserved through power failures. This bit is still asserted when the global SCI_EN is set to '1' in which case a System Control Interrupt (SCI) will be generated. The BIOS or SCI handler software clears this bit by writing a 1 to it.

The SoC also indicates a THERMTRIP_N event via the active-low THERMTRIP_N output signal pin. When CLTT is enabled for Memory Thermal Management (Section 35.5.1.2, "Close Loop Thermal Throttling (CLTT)"), the MEM_THERMTRIP can be configured to generate the THERMTRIP event.

Overall power to the SoC is eliminated fully once the VRs on the motherboard are shut down within milliseconds. The platform must remove all power (except battery power) to the SoC within 500 ms of a THERMTRIP_N signal assertion. A cold reset is required to recover from a THERMTRIP_N event.

## 35.4.6 Thermal Status and Interrupts

### 35.4.6.1 Thermal Status and Interrupts — Processor

The Operating System (OS) can manage both Core processor and SoC package thermal conditions natively without relying on BIOS or other system board components. The SoC can be programmed to generate interrupts for various thermal events. These interrupts are typically handled either by the OS or System Management Mode (SMM) software.

The thermal interrupt Delivery Mode, Interrupt Mask and the Interrupt Vector for this interrupt, can be programmed through the Thermal Monitor Register entry in the local core's APIC Local Vector Table (LVT), or optional modes may be configured that allow the thermal interrupt notification to be routed to a master core for processing. For more information on the APIC LVT Thermal Monitor Register, see Volume 3, Section 10.5 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

There are two programmable thermal thresholds per core plus two per package. There is no direct mapping to the number of DTS instances, as interrupts are implemented by the SoC power management mechanism, not in the DTS.

The per-core Thermal Status and per-core Thermal Interrupt use the temperature value of the DTS located near the shared L2-cache. This is important to know when accessing the Model-Specific Registers (MSRs).

The SoC Package Thermal Status and Interrupt uses the Maximum Package Temperature value which is defined as the maximum temperature of the 13 SoC DTSes. All DTS sensors are simultaneously monitored for catastrophic thermal conditions. If such a condition occurs, the CPU will perform an immediate shutdown.

**THERM_INTERRUPT**
IA32_THERM_INTERRUPT (MSR 19Bh)

There are six processor Thermal Interrupts that can be enabled:

- HIGH_TEMP_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon the rising edge of the Thermal Monitor Status of this core as it crosses the Thermal Monitor Activation Temperature to a temperature above it.
- LOW_TEMP_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon the falling edge of the Thermal Monitor Status of this core as it crosses the Thermal Monitor Activation Temperature to a temperature below it.
- PROCHOT_INT_ENABLE - Assertion or de-assertion of PROCHOT_N.
- OUT_OF_SPEC_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation that the processor is operating out of its thermal specification. This interrupt is designed as an early warning of thermal runaway in the silicon and shutdown is recommended.
- THRESHOLD_1_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation of any rising or falling detection of the core temperature crossing the THRESHOLD_1_REL_TEMP temperature.
- THRESHOLD_2_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation of any rising or falling detection of the core temperature crossing the THRESHOLD_2_REL_TEMP temperature.
- THRESHOLD_1_REL_TEMP and THRESHOLD_2_REL_TEMP - Each is a seven-bit value that represents a negative offset from the reference maximum junction temperature defined in TEMPERATURE_TARGET.REF_TEMP. These temperature thresholds are compared against the corresponding Thermal Status Relative Temperature reading for this core.

**THERM_STATUS**
IA32_THERM_STATUS (MSR 19Ch)
Per-Core DTS Temperature Read Data (PECI RdPkgConfig(), Index 9)

This register is typically used to inspect the thermal status of a specific core as well as to discover the sources of core thermal-interrupt events. There are seven types of processor Thermal Status each with a sticky monitor bit (Log bit) to let software know if a new status bit has been posted. Each status bit is Read-Only.

- POWER_LIMITATION_STATUS - Indicates that the processors are performance limited due to power constraints.
- THERMAL_MONITOR_STATUS - Indicates that any of the Thermal Monitors has tripped and that the SoC processors are currently thermally throttling.
- PROCHOT_STATUS - Indicates PROCHOT is currently asserted.
- OUT_OF_SPEC_STATUS - Indicates that a processor is operating out of its thermal specification.
- THRESHOLD1_STATUS - Indicates that the actual processor temperature is currently higher than or equal to the value set in Thermal Threshold 1.
- THRESHOLD2_STATUS - Indicates that the actual processor temperature is currently higher than or equal to the value set in Thermal Threshold 2.
- TEMPERATURE - Processor DTS temperature (DTS near shared L2 cache) reported as a negative offset in °C from the factory-set TEMPERATURE_TARGET.REF_TEMP.

*Note:* Because TEMPERATURE and the two temperature threshold values shown above are defined as negative offsets, a higher number means a lower temperature.

*Note:* The GetTemp() command via the PECI SMBus returns a temperature value which is different than the TEMPERATURE field shown here. See Chapter 30, “GetTemp()” and Chapter 30, “DTS Temperature Data.”

### 35.4.6.2 Thermal Status and Interrupts — SoC Package

The SoC package thermal status and interrupt are reported the same way the core status and interrupt is reported. See Section 35.4.6.1, "Thermal Status and Interrupts — Processor." The Thermal Monitor Register entry in the core APIC Local Vector Table (LVT) for package-scoped interrupts is identical to entries of core-scoped interrupts. Software is responsible for determining the origin of the interrupt.

**PACKAGE_THERM_INTERRUPT**
IA32_PACKAGE_THERM_INTERRUPT (MSR 1B2h)

There are seven package-level Thermal Interrupts that can be enabled:

- POWER_INT_ENABLE - When this bit is set a thermal interrupt will be sent upon detection of a rising-edge assertion of the Power Limit Status bit in the IA32_PACKAGE_THERM_STATUS (MSR 1B1h) register.
- HIGH_TEMP_INT_ENABLE - Rising edge of the Thermal Monitor Status of the virtual maximum package temperature.
- LOW_TEMP_INT_ENABLE - Falling edge of the Thermal Monitor Status of the virtual maximum package temperature.
- OUT_OF_SPEC_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation that a processor or un-core DTS is operating out of its thermal specification. This interrupt is designed as an early warning of thermal runaway in the silicon and shutdown is recommended.
- PROCHOT_INT_ENABLE - Assertion or de-assertion of PROCHOT_N.
- THRESHOLD_1_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation of any rising or falling detection of the SoC package maximum temperature crossing the Threshold 1 temperature.
- THRESHOLD_2_INT_ENABLE - This bit allows software to enable thermal interrupt delivery upon observation of any rising or falling detection of the SoC package maximum temperature crossing the Threshold 2 temperature.

Threshold 1 and Threshold 2 are 7-bit values in IA32_PACKAGE_THERM_INTERRUPT that represent a negative offset from the reference maximum junction temperature defined in MSR_TEMPERATURE_TARGET.REF_TEMP. These temperature thresholds are compared against the corresponding Thermal Status Relative Temperature reading for the "virtual package max temperature." The possible interrupts and associated enable bis are mentioned above.

**PACKAGE_THERM_STATUS**

IA32_PACKAGE_THERM_STATUS (MSR 1B1h)
Package Thermal Status Data (PECI RdPkgConfig(), Index 20)

There are seven types of package-level Thermal Status, each with a sticky monitor bit (Log bit) to let software know if a new status bit has been posted. Each status bit is Read-Only.

- POWER_LIMITATION_STATUS - Indicates that the processors are performance limited due to power constraints.
- THERMAL_MONITOR_STATUS - Indicates that any of the package Thermal Monitors has tripped and that the package is currently thermally throttling.
- PROCHOT_STATUS - Indicates PROCHOT is currently asserted.
- OUT_OF_SPEC_STATUS - Indicates that a processor is operating out of its thermal specification.
- THRESHOLD1_STATUS - Indicates that the actual package temperature is currently higher than or equal to the value set in Thermal Threshold 1.
- THRESHOLD2_STATUS - Indicates that the actual package temperature is currently higher than or equal to the value set in Thermal Threshold 2.
- TEMPERATURE - Package temperature reported as a negative offset in °C from the factory-set TEMPERATURE_TARGET.REF_TEMP.

*Note:* Because TEMPERATURE and the two temperature thresholds shown above are defined as negative offsets, a higher number means a lower temperature.

## 35.5 Memory Thermal Management (MTM)

The SoC can provide thermal management of the DIMM components of the platform board that are connected to the SoC Memory Controllers. After the SoC memory subsystem has been setup by BIOS the BIOS needs to indicate how memory thermals are to be managed by the SoC. The SoC-internal registers written by BIOS at boot time must remain statically configured.

*Note:* The Memory Thermal Management registers are not accessible as SoC Model-Specific Registers (MSRs).

### 35.5.1 Thermal Throttling

Three Memory Thermal Management (MTM) methods are available: Open Loop Thermal Throttling (OLTT); Closed Loop Thermal Throttling (CLTT); CLTT Pass-Through. The MTM mode is selected via the P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR register. Memory throttling can be configured to generate an SMI.

There are no Model-Specific Registers (MSRs) associated with MTM.

The SoC does not support the reading of the DRAM device's Mode Register 4 (MR4). Because of this, memory subsystems that do not read the DIMM temperature sensors via Serial Presence Detect (SPD) and the SoC Legacy SMBus, or have the ability to provide the SoC the DIMM temperature via the PECI SMBus, need to use OLTT for Memory Thermal Management.

The SoC has a dedicated Memory Hot input indicator pin, named MEMHOT_N, to allow the platform to indicate when memory thermal throttling is required. It is a fast-responding event to the SoC power-management control, approximately 100 µs. The amount of thermal throttling can be programmed during the setup at boot time.

The SoC response to the MEMHOT_N input signal pin can be enabled and configured in the P_CR_MEMHOT_THERM_CONFIG_0_0_0_MCHBAR register. The three bandwidth throttle levels of P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR are used in the P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR register.

The state of the MEMHOT_N signal pin is provided in the P_CR_MEMHOT_THERM_STATUS_0_0_0_MCHBAR register.

#### 35.5.1.1 Open Loop Thermal Throttling (OLTT)

The OLTT mode provides a fixed level of memory thermal throttle via a bandwidth limit mechanism programmed by BIOS during boot time. The bandwidth throttle level used is programmable in the two-bit OLTT_THROT_LVL field of the P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR register. The throttle level values are the same as the values programmed for CLTT mode and explained in Section 35.5.1.3, "CLTT Throttle Levels." The throttle-level values can be altered at runtime.

The memory bandwidth limits work by limiting the number of memory reads/writes within a given interval. BIOS programs the bandwidth percentage and the SoC converts that bandwidth percentage into the number of reads and writes for a particular interval.

*Note:* Even though the P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR register is associated with CLTT mode, its three throttling level values are used for OLTT mode and for the MEMHOT_N input signal pin mechanism.

### 35.5.1.2 Close Loop Thermal Throttling (CLTT)

In CLTT mode the SoC provides three configurable levels of memory bandwidth throttling that are triggered in response to temperature data sent from the DIMMs to the SoC. The SoC can be programmed to use one of two ways to receive this temperature data:

- Normal: DIMM temperature readings are periodically obtained via SPD by the SoC using the Legacy SMBus interface.
- Pass-through: Temperature readings are provided to the SoC from the platform Baseboard Management Controller (BMC) via the SMBus PECI interface.

DIMM temperature data are provided by a “Temperature Sensor On DIMM” (TSOD).

For normal CLTT the temperature readings are provided directly from the Legacy SMBus interface.

For pass-through CLTT the BMC is required to acquire TSOD temperature measurements from the DIMM or obtain DIMM temperature information by some other means or mechanism, and write the measurements to the SoC using PECI commands via the SoC PECI SMBus. The BMC, or other means is expected to update the DIMM temperature values via the PECI SMBus every 5-10 ms.

The SoC supports dynamic CLTT which allows the platform to make run-time updates to MTM configuration values that adjusts the temperature dynamic range in response to memory subsystem environmental changes detected by the platform.

### 35.5.1.3 CLTT Throttle Levels

In CLTT normal mode the MTM throttle level is based on the threshold crossed by the highest, adjusted DIMM temperature value the SoC receives via SPD over the SoC Legacy SMBus interface. The throttle levels are programmed by BIOS in the P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR register:

00 = no throttle
01 = THRT_MID (default is 0% throttled)
10 = THRT_HI (default is 90% throttled)
11 = THRT_CRIT (default is 99% throttled)

The % values of throttling are interpreted as shown in the following example:

If the value is 25% throttled then 75% of the maximum memory bandwidth will be available.

In CLTT mode, the throttle-level values in the P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR register can be adjusted after BIOS boot time.

The memory bandwidth limits work by limiting the number of memory reads/writes within a given interval. BIOS programs the bandwidth percentage, and the SoC converts that bandwidth percentage into the number of reads and writes for a particular interval.

### 35.5.1.4 CLTT Temperature Threshold Values

In CLTT mode, there are three programmable temperature threshold values which trigger MTM responses. They are programmed by BIOS. Their names, location in the P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR register, default values, and MTM response are:

- TEMP_LO (bit 1), TSOD = 84 °C, Initiate 2x refresh rate and THRT_MID.
- TEMP_MID (bit 10), TSOD= 93 °C, Initiate THRT_HI, continue 2x refresh rate.
- TEMP_HI (bit 11), TSOD = 100 °C, Initiate THRT_CRIT, continue 2x refresh rate, assert MEM_THERMTRIP.

The P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR register also contains a three-bit TEMP_THROT_HYST field that provides a programmable, negative-going threshold hysteresis value that is programmable by BIOS.

To account for offsets between DIMM hot-spot temperature and the reported TSOD values, per-DIMM temperature offsets can be programmed in the P_CR_MEM_THERM_TEMP_OFST_0_0_0_MCHBAR register.

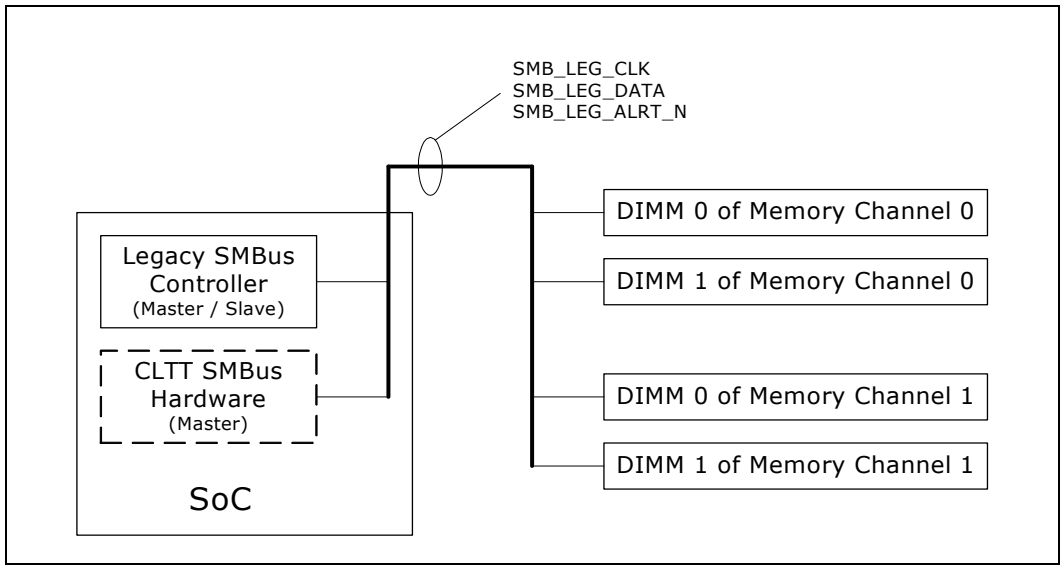
*Note:* It is the responsibility of the platform to ensure that current, accurate temperature data is provided. The SoC Memory Thermal Management algorithms have no mechanism to detect if temperature data is stale or invalid. If TSODs malfunction, it is recommended that the platform utilize the MEMHOT_N input signal pin as a backup mechanism to control DIMM temperature.

The offset temperature values in the P_CR_MEM_THERM_TEMP_OFST_0_0_0_MCHBAR register and the throttle thresholds in the P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR register that the SoC uses to create variable MTM throttle thresholds, can be based on system information such as airflow. These registers can be written by the BMC via the PECI WrEndPointConfig() command over the SoC PECI SMBus interface. This PECI command writes the appropriate register located within the SoC sideband registers. See Chapter 30, "WrEndPointConfig()."

### 35.5.1.5 Legacy SMBus Used for Accessing Information for CLTT

When in CLTT Normal mode, the SoC uses a special internal connection to the Legacy SMBus interface as an SMBus Master to acquire DIMM-related CLTT information from the system DIMMs. See Figure 35-2, "CLTT SMBus Connection to DIMMs."

**Figure 35-2. CLTT SMBus Connection to DIMMs**




SoC commands to the SMBus slave (one of the installed DIMMs) are initiated through SoC-internal messages to the SoC SMBus interface hardware by the SoC power-management controller. As an SMBus bus master, the CLTT-based SoC SMBus hardware issues the particular transaction on the SMBus. Only a limited set of SMBus transactions are supported for this purpose.

Once the transaction over the SMBus interface has been completed, the SoC SMBus hardware places the data in SoC-internal registers and responds to the SoC power-management controller that the SMBus operation has been completed.

The CLTT SMBus operations do not support the SMBus Alert option (SMB_LEG_ALRT_N signal). The Legacy SMBus controller does support SMBus Alert.

The Temperature Sensor On DIMM (TSOD) information acquired by this SMBus mechanism is available to software and BIOS through the SoC Sideband registers:

- C0TEMPSTS0 - DIMM 0 of Memory Channel 0
- C0TEMPSTS1 - DIMM 1 of Memory Channel 0
- C1TEMPSTS0 - DIMM 0 of Memory Channel 1
- C1TEMPSTS1 - DIMM 1 of Memory Channel 1

### 35.5.1.6 PECI SMBus Used for Accessing Information for CLTT

When in CLTT Pass-Through mode, the BMC or some other external management agent writes temperature measurements periodically to the SoC using a PECI write command via the SoC PECI SMBus interface. See Chapter 30, "DRAM Rank Temperature Write (Index = 18)" in the PECI SMBus chapter.

### 35.5.2 MEM_THERMTRIP

Internal to the SoC, MEM_THERMTRIP indicates that a DIMM temperature has exceeded the highest temperature threshold and memory is considered to be in a state of catastrophic thermal failure. The MEM_THERMTRIP state is determined by the Memory Thermal Management (MTM) throttling algorithm but initiates an additional configurable response that can trigger a processor THERMTRIP_N event.

The SoC-generated P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR register bits are used by the SoC to generate and internal MEM_THERMTRIP indication.

If configured to do so, MEM_THERMTRIP can generate the SoC-internal THERMTRIP event indication and assert the SoC THERMTRIP_N signal pin. This is configured by the EN_MEMTRIP field of the P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR register.

### 35.5.3 Thermal Status and Interrupts

While the SoC does not generate interrupts for Memory Thermal Management events, it does provide various CLTT and customer-configurable status and logs available through the P_CR_MEM_THERM_INT_STATUS_0_0_0_MCHBAR register:

- "CLTT SMBus is hung" status and log
- "CLTT SMBus is reset" status and log
- DIMM temperature is at or above the point to cause SoC memory throttling. Status and log.

Each log is one bit that is set to 1 the first time the particular status is set and remains sticky until cleared via software or by reset.

When in CLTT mode, either in Normal or Pass-Through mode, the SoC provides thermal status and a log for each of the DIMMs of the memory subsystem. These can be read by software in the P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR register. Each DIMM has a two-bit field:

- 00 = Memory temperature < TEMP_LO
- 01 = TEMP_LO ≤ Memory temperature < TEMP_MID
- 10 = TEMP_MID ≤ Memory temperature < TEMP_HI
- 11 = Memory temperature ≥ TEMP_HI (MEM_THERMTRIP condition)

For TEMP_HI, TEMP_MID, and TEMP_LO, see Section 35.5.1.4, "CLTT Temperature Threshold Values."

Each DIMM has a two-bit Thermal Throttle log in the P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR register. The logs are sticky bits set to highest throttle level achieved (same encoding as P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR register fields) when CLTT memory throttling is initiated by the particular DIMM. The logs remain set until cleared when software writes a 1 value to the log field or the SoC is reset. For more information about memory throttle levels, see Section 35.5.1.3, "CLTT Throttle Levels."

## 35.6 Run Average Power Limit (RAPL)

RAPL provides platform software the ability to monitor, control, and get notifications about SoC power consumption. The SoC provides RAPL for managing platform thermals.

The Thermal Design Power (TDP) specified for the SoC represents the maximum amount of power the platform cooling system required to dissipate in order for the SoC to operate. But it is possible for the processor to consume more than the TDP power for a short period of time without it being thermally significant. The primary use case of RAPL is providing software the ability of setting per-package, hardware-enforced, power limits. In addition, RAPL provides a very accurate power meter capable of supporting core and non-core devices.

The SoC supports two RAPL domains of power rationing:

- SoC Package domain which includes everything on the die
- DRAM domain which includes the directly-attached DRAM

The PKG_POWER_LIMIT and DRAM_POWER_LIMIT registers are used to configure which RAPL domain is to be enabled. Having both SoC Package RAPL and DRAM RAPL combined into a single RAPL limit called Super-Domain RAPL is not supported.

The SoC supports external access to the RAPL domain interfaces three ways:

- Non-architectural Model-Specific Registers (MSRs) using RDMSR and WRMSR instructions
- Memory-Mapped I/O (MMIO) registers in Host Root Space
- By the BMC, or equivalent, through PECI SMBus commands to access the RAPL registers.

The following set of capabilities are supported through the interfaces:

- Power Limit - Specify power limit, time window, and other controls. RAPL provides a way to set short-term and longer-term averaging windows for Power Limits. Platform thermal control software is advised to dynamically update the RAPL Power Limits and the averaging windows to meet SoC system requirements.
- Energy Status -Access the power-metering interface providing energy consumption information.
- Perf Status - Access to information on the performance effects (regression) due to Power Limits. It is defined as a duration metric that measures the Power Limit effect in the respective domain.
- Power Info - Interface providing information on the range of parameters for a given domain.

The SoC provides the Read-Only RAPL_POWER_UNIT register for software to discover the basic units of time, energy and power defined for SoC Package RAPL and DRAM RAPL.

For additional details about RAPL, see Volume 3, Chapter 14 of the *Intel® 64 and IA-32 Architectures Software Developer's Manual*.

### 35.6.1 RAPL — SoC Package

For SoC Package RAPL, both Power Limit 1 (PL1) and PL2 Power/Time tuples are supported.

PL1 is typically used for long thermal-time constants, defaulting to Thermal Design Power (TDP) Power/Time. PL2 is typically programmed for a shorter Power/Time tuple to manage Dynamic Turbo where power is allowed to exceed TDP for short, thermal time constants, thus allowing burst performance. To mitigate power, the SoC manages internal power-management mechanisms if required to meet the power tuples. Both PL1 and PL2 have their own control systems which operate concurrently.

The SoC also provides what is called “Fast RAPL” which is a special duty-cycle protection to control excursions past PL2 to limit the duty-cycle of power transients greater than PL2. The purpose of this protection is to avoid thermal over-design of the platform Voltage Regulator (VR). The protection allows the guaranteed VR duty-cycle specifications to be greater than PL2 alone would allow. The SoC uses the SVID interface to read the output current of VCC, VNN and VCCRAM VRs to calculate power to provide this special protection. The VR needs to be capable of being sampled at 100-µs intervals for output current as well using its internal A/D converter to provide a 200-µs anti-aliasing filter prior to digital conversion.

The registers associated with SoC Package RAPL are accessible by RDMSR/WRMSR instructions, through Memory-Mapped I/O (MMIO), or using PECI commands via the PECI SMBus interface. In some cases the PECI information has a different format:

**RAPL_POWER_UNIT**

MSR_RAPL_POWER_UNIT (MSR 606h)
P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR (MMIO MCHBAR + 7068h)
Package Power SKU Unit Read Data (PECI RdPkgConfig(), Index 30)

This register defines the units for calculating all SoC Package RAPL and DRAM RAPL RAPL power, current, energy, resistance and timing parameters.

**PKG_POWER_LIMIT**

MSR_PKG_POWER_LIMIT (MSR 610h)
P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR (MMIO MCHBAR + 70A8h)
Package Power Limits for Multiple Turbo Modes (PECI WrPkgConfig(), Index 26)
Package Power Limits for Multiple Turbo Modes (PECI RdPkgConfig(), Index 27)

This register allows software to define power limitation for the SoC Package RAPL domain. PL1 and PL2 power limits and associated time windows can be specified. The values are expressed in units defined by the RAPL_POWER_UNIT register.

**PKG_ENERGY_STATUS**

MSR_PKG_ENERGY_STATUS (MSR 611h)
P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 706Ch)
Accumulated Energy Status Read Data (PECI RdPkgConfig(), Index 03)

This 32-bit Read-Only register contains an accumulated value of the energy consumed by the entire SoC package. The value of this register is updated by the SoC approximately once every 1 ms. The units are proportional to Joules and are defined by the RAPL_POWER_UNIT register. The SoC uses this value for PL1 and PL2 RAPL algorithms.

**PKG_PERF_STATUS**

MSR_PKG_PERF_STATUS (MSR 613h)
P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 7050h)
Package Power Limit Performance Status Read Data (PECI RdPkgConfig(), Index 08)

Measures the time that any core in the SoC package is performance-throttled below what the Operating System requested and below the core's base frequency because of Power Limits PL1 or PL2. The time value is expressed in units defined by the RAPL_POWER_UNIT register.

**PKG_POWER_INFO**

MSR_PKG_POWER_INFO (MSR 614h)
P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR (MMIO MCHBAR + 70A0h)
Package Power SKU Read Data (PECI RdPkgConfig(), Index 28 and 29)

This register provides the factory-defined limits for the configurable PL1 and PL2 SoC Package Power Limits and Time Windows in the PKG_POWER_LIMIT register. The values are expressed in units defined by the RAPL_POWER_UNIT register.

**PP0_ENERGY_STATUS**

MSR_PP0_ENERGY_STATUS (MSR 639h)
P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 705Ch)

This register contains an accumulated value of the energy consumed in the primary power plane across all cores of the SoC. The value is expressed in units defined by the RAPL_POWER_UNIT register. The value of this register is updated approximately once every 1 ms.

**PP1_ENERGY_STATUS**

MSR_PP1_ENERGY_STATUS (MSR 641h)
P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 7060h)

This register contains an accumulated value of the energy consumed in the secondary power plane across all cores of the SoC. The value is expressed in units defined by the RAPL_POWER_UNIT register. The value of this register is updated approximately once every 1 ms.

**TURBO_ACTIVATION_RATIO**

MSR_TURBO_ACTIVATION_RATIO (MSR 64Ch)

P_CR_TURBO_ACTIVATION_RATIO_0_0_0_MCHBAR (MMIO MCHBAR + 70F0h)

ACPI P-T Notify Write Data (PECI WrPkgConfig(), Index 33)
ACPI P-T Notify Read Data (PECI RdPkgConfig(), Index 33)

This register is used by software to configure the Performance State (P-State) at which Turbo activates. Turbo activation starts at clock ratios above the eight-bit Turbo Activation Ratio Threshold (MAX_NON_TURBO_RATIO field) configured in this register. This feature is disabled when the threshold is set to zero.

**IA_PERF_LIMIT_REASONS**

MSR_IA_PERF_LIMIT_REASONS (MSR 64Fh)

P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR (MMIO MCHBAR + 70B0h)

This register reports reasons for performance limitations on the processor cores. Status bits are an instantaneous indication of an active constraint. Log bits indicate that a constraint was enforced since the log bit was last cleared.

**CORE_PERF_LIMIT_REASONS**

MSR_CORE_PERF_LIMIT_REASONS (MSR 690h)

Indicator of Frequency Clipping in Processor Cores (R/W). Frequency refers to processor core frequency. I see 11 status fields and 11 associated logs.

**PL3_CONTROL**

*Official MSR Name is TBD* (MSR 615h)
P_CR_PL3_CONTROL (Sideband Port P47h, offset 4790h)

Control Power Limit 3 (PL3) and Power Limit 4 (PL4) using this register.

This limit control is physically different from the same control in the IA core MSR space.

— PL3 is designed to clamp peak sustained power to levels supported by the battery or input power supply and as such manage lifetime degradation of that power delivery element. With PL3, peak power excursions above the limit are allowed so long as they do not exceed the configured duty cycle constraint in this register.

— PL4 is designed to clamp peak instantaneous power to levels below the max supported by the battery or input power supply. These clamps are implemented a priori and the SoC is guaranteed to constrain itself below the PL4 limit always.

## 35.6.2 RAPL — DRAM

*Note:* DRAM RAPL is not supported when Non-Volatile DIMMs (NVDIMMs) are used.

*Note:* Both Memory Thermal Management (MTM) and RAPL are capable of throttling memory bandwidth. When both MTM and RAPL are in effect, the memory bandwidth limit will be set to the more restrictive limit between MTM and RAPL. When either condition (MTM or RAPL) is removed, the remaining memory bandwidth limit will be invoked until conditions change such that a different limit, or no limit, is required.

DRAM RAPL is a SoC feature that manages the DRAM power budget to a programmable value. The SoC only supports DRAM RAPL Power Level 1 (PL1) Limit Power/Time tuple. RAPL manages the DRAM domain power which comprises the summation of all the DIMMs populated on the two Memory Controller channels: Two DIMMs per channel; Two ranks per DIMM. To keep within the set Power/Window values, the DIMM power consumption is mitigated by the SoC controlling the memory-access bandwidth by using bandwidth throttling.

The registers associated with DRAM RAPL are accessible by either RDMSR/WRMSR instructions, through Memory-Mapped I/O (MMIO), or using PECI commands via the PECI SMBus interface. In some cases the PECI information has a different format:

**DRAM_POWER_LIMIT**
MSR_DRAM_POWER_LIMIT (MSR 618h)
P_CR_DDR_RAPL_LIMIT_0_0_0_MCHBAR (MMIO MCHBAR + 7040h)
DRAM Power Limit Data (PECI RdPkgConfig()/WrPkgConfig(), Index 34)

This register is written by the platform software and read by the SoC approximately once every 1 ms. It contains the DRAM Power Limit enable, the DRAM Power Limit value, and the Time Window value. The values are expressed in units defined by the RAPL_POWER_UNIT register.

**DRAM_ENERGY_STATUS**
MSR_DRAM_ENERGY_STATUS (MSR 619h)
P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 7048h)
Accumulated DRAM Energy Read Data (PECI RdPkgConfig(), Index 04)

This 32-bit Read-Only register contains the Joules of energy consumed by all DIMMs. The value of this register is updated by the SoC approximately once every 1 ms. The units are proportional to Joules and are defined by the RAPL_POWER_UNIT register.

**DRAM_PERF_STATUS**
MSR_DRAM_PERF_STATUS (MSR 61Bh)
P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR (MMIO MCHBAR + 704Ch)
DRAM Power Limit Performance Status Read Data (PECI RdPkgConfig(), Index 38)

Memory RAPL performance throttling counter. This counter accumulates time that any channel in the memory controller is bandwidth-throttled due to memory RAPL constraints. The time values are expressed in units defined by the RAPL_POWER_UNIT register.

**DRAM_POWER_INFO**
MSR_DRAM_POWER_INFO (MSR 61Ch)
P_CR_DRAM_POWER_INFO_0_0_0_MCHBAR (MMIO MCHBAR + 7130h)
DRAM Power Info Read Data (PECI RdPkgConfig(), Index 35 and 36)

Various DRAM RAPL power and time-window information.

## 35.7 Inverse Temperature Dependence (ITD)

Inverse Temperature Dependence (ITD) is a SoC feature where internal circuitry requests the platform board VRs to increase SoC voltages at cold temperatures to compensate for timing-path slowdown when cold. The ITD feature is provided on all SoC product SKUs for the VCC, VCCSRAM, and VNN voltage groups. The temperature values and voltage offsets are set at the Intel factory.

*Note:* ITD is also known as Low Temperature Effect Compensation (LTEC).

## 35.8 Platform Voltage Regulator Thermal Alert

The SoC supports VR_THERMALERT. The SoC polls the SVID status message for the bit that indicates the platform board Voltage Regulator (VR) is approaching its maximum operating temperature. The VR asserts the VR_THERMALERT bit in the SVID message at a VR temperature lower than the one it uses to assert its VR-hot# pin. The pin is used to drive the SoC PROCHOT_N signal pin if PROCHOT_N is configured as a SoC input.

In response to the VR_THERMALERT indication the SoC reduces all cores in Turbo Mode (P0 Performance State) directly to the P1 Performance State, and clamps it there, but does not throttle (lower the SoC voltage and clock-frequency) any further. Cores in lower P-states remain in those states. No interrupt is generated.

Turbo is then disallowed until the VR_THERMALERT condition is removed.

The VR_THERMALERT signaling is disabled when VR_THERM_ALERT_DISABLE, bit 24 of the MSR_POWER_CTL register (MSR 1FCh), is set to 1.

There are three ways software can access the access the VR_THERMALERT status:

**PERF_LIMIT_REASONS**

MSR_IA_PERF_LIMIT_REASONS (MSR 64Fh)

P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR (MMIO MCHBAR + 70B0h)

The fields are:

VR_THERMALERT_STATUS (bit 10) - Read-Only. Clock frequency is limited due to a VR thermal excursion.

VR_THERMALERT_LOG (bit 26) - Logged indication that frequency was clamped due to a VR thermal excursion. This bit is set by the SoC and can be cleared by software.

## 35.9 Model-Specific Registers (MSR)

**Table 35-4. MSR Registers**

| MSR Address (hex) | Register Name | Register Name (Basic Name) |
|---|---|---|
| 199 | IA32_PERF_CTL | PERF_CTL |
| 19A | IA32_CLOCK_MODULATION | CLOCK_MODULATION |
| 19B | IA32_THERM_INTERRUPT | THERM_INTERRUPT |
| 19C | IA32_THERM_STATUS | THERM_STATUS |
| 1A0 | IA32_MISC_ENABLE | MISC_ENABLE |
| 1A1 | MSR_PLATFORM_BRV | PLATFORM_BRV |
| 1A2 | MSR_TEMPERATURE_TARGET | TEMPERATURE_TARGET |
| 1AA | MSR_MISC_PWR_MGMT | MISC_PWR_MGMT |
| 1B1 | IA32_PACKAGE_THERM_STATUS | PACKAGE_THERM_STATUS |
| 1B2 | IA32_PACKAGE_THERM_INTERRUPT | PACKAGE_THERM_INTERRUPT |
| 1FC | MSR_POWER_CTL | POWER_CTL |
| 606 | MSR_RAPL_POWER_UNIT | RAPL_POWER_UNIT |
| 610 | MSR_PKG_POWER_LIMIT | PKG_POWER_LIMIT |
| 611 | MSR_PKG_ENERGY_STATUS | PKG_ENERGY_STATUS |
| 613 | MSR_PKG_PERF_STATUS | PKG_PERF_STATUS |
| 614 | MSR_PKG_POWER_INFO | PKG_POWER_INFO |
| 618 | MSR_DRAM_POWER_LIMIT | DRAM_POWER_LIMIT |
| 619 | MSR_DRAM_ENERGY_STATUS | DRAM_ENERGY_STATUS |
| 61B | MSR_DRAM_PERF_STATUS | DRAM_PERF_STATUS |
| 61C | MSR_DRAM_POWER_INFO | DRAM_POWER_INFO |
| 64F | MSR_IA_PERF_LIMIT_REASONS | IA_PERF_LIMIT_REASONS |

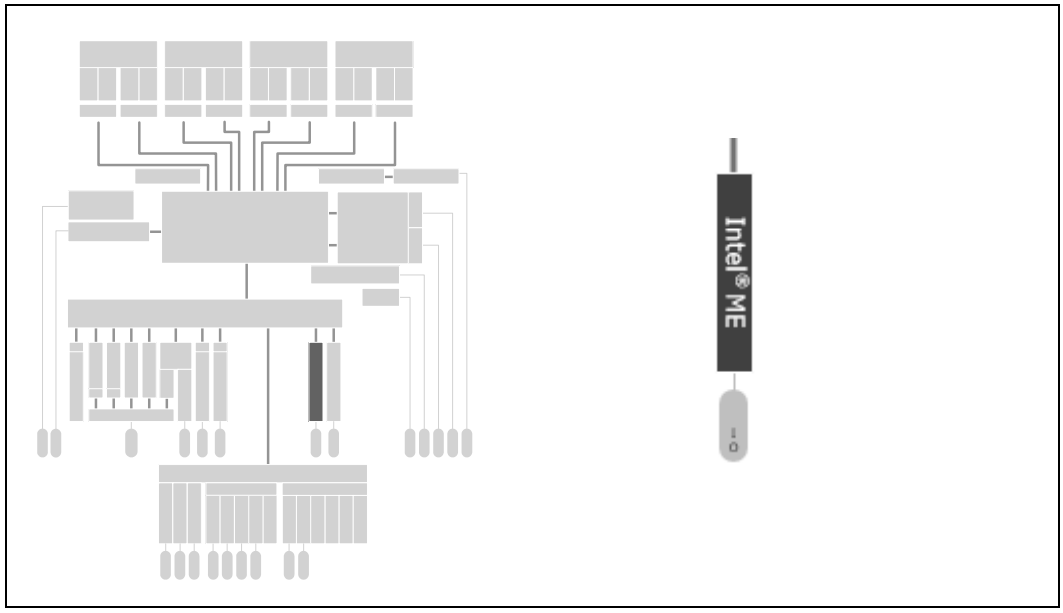
§ §

# 36 System Management

The Intel® Management Engine (Intel® ME) is a subsystem that provides the system management. The Intel® ME firmware is provided and developed by Intel.

The Innovation Engine (IE) is an optional feature through which SoC customers can create a custom system-management subsystem. See Chapter 37, "Innovation Engine."

**Figure 36-1. What is Covered in This Chapter**



## 36.1 Signal Descriptions

The Intel® ME has three SMBus controllers that tie into the SoC SMBus interfaces. The signals for these SMBus controllers are shown in Chapter 27, "System Management Bus (SMBus) Controller - Overview."

For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing." The Direction/Type column of Table 36-1, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 36-1. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| SMB_ME_SMT0_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 0 of the Intel® ME |
| SMB_ME_SMT0_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 0 of the Intel® ME. |
| SMB_ME_SMT0_ALRT_N | I,O-OD | Yes | **SMBus Alert:** (SMBALERT#) for System Management Transport 0 of the Intel® ME. |
| SMB_ME_SMT1_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 1 of the Intel® ME. |
| SMB_ME_SMT1_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 1 of the Intel® ME. |
| SMB_ME_SMT2_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 2 of the Intel® ME. |
| SMB_ME_SMT2_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 2 of the Intel® ME. |

## 36.2 Intel® ME and Intel Server Platform Services Firmware

### 36.2.1 Key Properties of Intel® ME

- Connectivity:
  - Delivery of advanced I/O functions.
- Security:
  - More secure and isolated execution.
- Modularity and Partitioning:
  - OSV, VMM and SW independence.
- Power:
  - When in S0, the SoC can be configured by BIOS to completely shut down Intel® ME power.
  - The SoC has a power-disable policy that can be set to shut down the Intel® ME and all its contents are lost.
  - Advanced functions in low power S5 operation.
  - OS independent PM and thermal heuristics.

### 36.2.2 Intel® ME Requirements

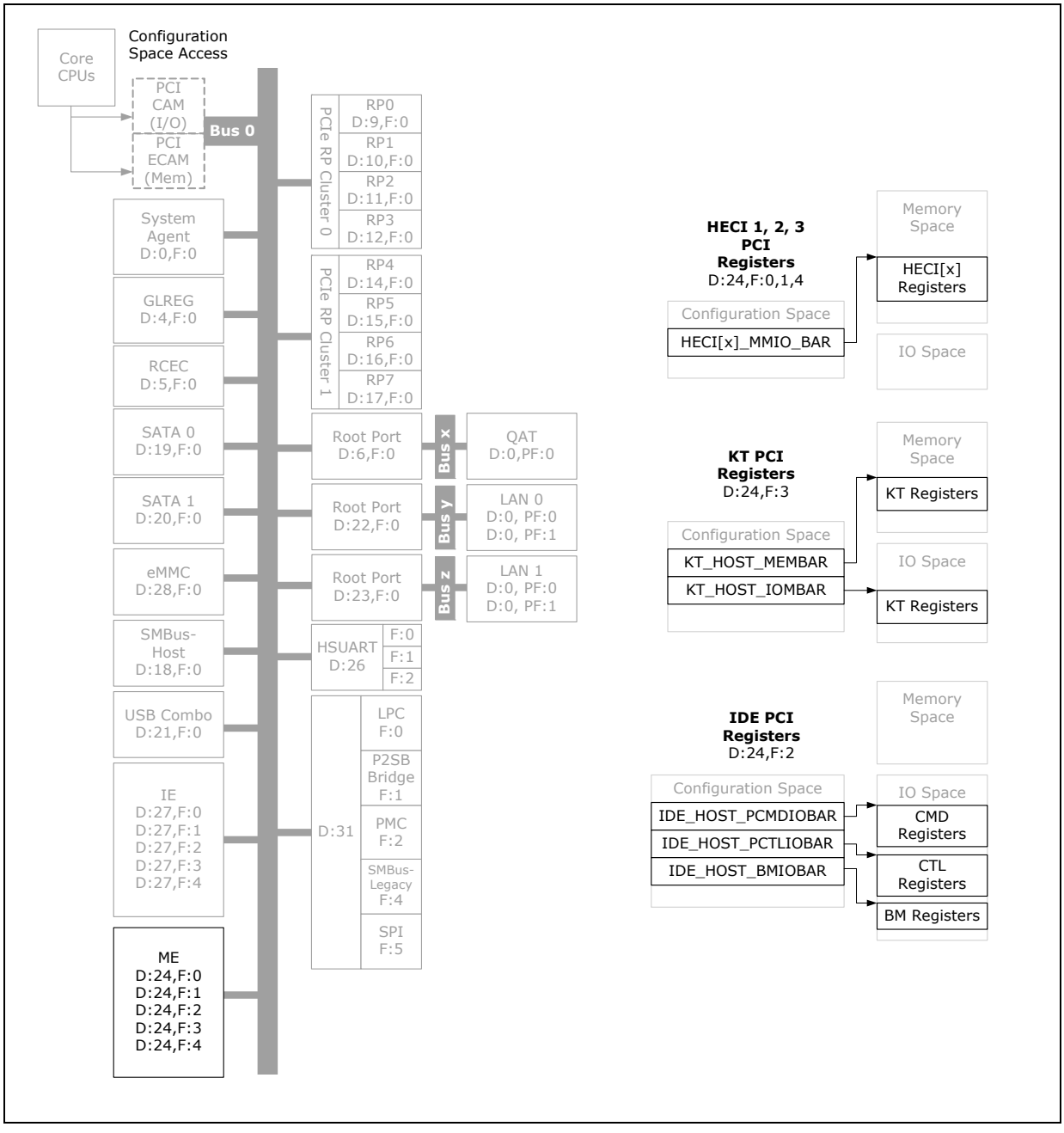
The following list of components composes the Intel® ME hardware infrastructure.

- The Intel® ME is the general purpose controller that resides in the SoC. It operates in parallel to, and is resource-isolated from, the host processor.
- The SPI flash device stores Intel Server Platform Services (SPS) FW code that is executed by the Intel® ME for its operations. The SoC controls the flash device through the SPI interface and internal logic.
- The Intel® ME interacts with the SoC BIOS/UEFI to make sure the Flexible I/O Adapter (FIA) and associated High-Speed I/O (HSIO) lanes are properly configured. See Chapter 10, “Flexible I/O Adapter (FIA) Overview.”

## 36.3 Register Map

The Intel® ME is accessible through the Host Root Space as shown in Figure 36-2, "Register Map."

**Figure 36-2. Register Map**



## 36.3.1 Host Root Space - Intel® ME HECI1

### 36.3.1.1 PCI Configuration and Capabilities

**Table 36-2. Configuration and Capabilities Register Map (Sheet 1 of 2)**

| Offset from Configuration Space of B0:D24:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI1_ID | Identifiers |
| 4 | HECI1_CMD | Command |
| 6 | HECI1_STS | Status |
| 8 | HECI1_RID_CC | Revision ID and Class Code |
| C | HECI1_CLS | Cache Line Size |
| D | HECI1_MLT | Master Latency Timer |
| E | HECI1_HTYPE | Header Type |
| F | HECI1_BIST | Built In Self-Test |
| 10 | HECI1_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI1_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI1_SS | Sub System Identifiers |
| 34 | HECI1_CAP | Capabilities Pointer |
| 3C | HECI1_INTR | Interrupt Information |
| 3E | HECI1_MGNT | Minimum Grant |
| 3F | HECI1_MLAT | Maximum Latency |
| 40 | HECI1_HFS | Host Firmware Status |
| 44 | HECI1_MISC_SHDW | Miscellaneous Shadow |
| 48 | HECI1_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI1_H_GS1 | Host General Status |
| 50 | HECI1_PID | PCI Power Management Capability ID |
| 52 | HECI1_PC | PCI Power Management Capabilities |
| 54 | HECI1_PMCS | PCI Power Management Control and Status |
| 60 | HECI1_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI1_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI1_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI1_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI1_H_GS2 | Host General Status 2 |
| 74 | HECI1_H_GS3 | Host General Status 3 |
| 8C | HECI1_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI1_MC | Message Signaled Interrupt Message Control |
| 90 | HECI1_MA | Message Signaled Interrupt Message Address |
| 94 | HECI1_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI1_MD | Message Signaled Interrupt Message Data |
| A0 | HECI1_HIDM | HECI Interrupt Delivery Mode |

## Table 36-2. Configuration and Capabilities Register Map (Sheet 2 of 2)

| Offset from Configuration Space of B0:D24:F0 (hex) | Register Short Name | Description |
|---|---|---|
| BC | HECI1_HERS | Host Extend Register Status |
| C0 | HECI1_HER1 | Host Extend Register DW1. |
| C4 | HECI1_HER2 | Host Extend Register DW2. |
| C8 | HECI1_HER3 | Host Extend Register DW3. |
| CC | HECI1_HER4 | Host Extend Register DW4. |
| D0 | HECI1_HER5 | Host Extend Register DW5. |
| D4 | HECI1_HER6 | Host Extend Register DW6. |
| D8 | HECI1_HER7 | Host Extend Register DW7. |
| DC | HECI1_HER8 | Host Extend Register DW8. |
| F8 | HECI1_MANID | Manufacturer's ID |

### 36.3.1.2 Memory-Mapped Registers

**Table 36-3. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI1_MMIO_MBAR (HI,LO) | 0 | HECI1_H_CB_WW | Host CB Write Window |
| HECI1_MMIO_MBAR (HI,LO) | 4 | HECI1_H_CSR | Host Control and Status Register |
| HECI1_MMIO_MBAR (HI,LO) | 8 | HECI1_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI1_MMIO_MBAR (HI,LO) | C | HECI1_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI1_MMIO_MBAR (HI,LO) | 800 | HECI1_D0I3C | D0i3 Control |

### 36.3.1.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 36.3.1.4 Sideband Registers

This device has no Sideband Registers.

## 36.3.2 Host Root Space - Intel® ME HECI2

### 36.3.2.1 PCI Configuration and Capabilities

**Table 36-4. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D24:F1 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI2_ID | Identifiers |
| 4 | HECI2_CMD | Command |
| 6 | HECI2_STS | Status |
| 8 | HECI2_RID_CC | Revision ID and Class Code |
| C | HECI2_CLS | Cache Line Size |
| D | HECI2_MLT | Master Latency Timer |
| E | HECI2_HTYPE | Header Type |
| F | HECI2_BIST | Built In Self-Test |
| 10 | HECI2_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI2_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI2_SS | Sub System Identifiers |
| 34 | HECI2_CAP | Capabilities Pointer |
| 3C | HECI2_INTR | Interrupt Information |
| 3E | HECI2_MGNT | Minimum Grant |
| 3F | HECI2_MLAT | Maximum Latency |
| 40 | HECI2_HFS | Host Firmware Status |
| 48 | HECI2_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI2_H_GS1 | Host General Status |
| 50 | HECI2_PID | PCI Power Management Capability ID |
| 52 | HECI2_PC | PCI Power Management Capabilities |
| 54 | HECI2_PMCS | PCI Power Management Control and Status |
| 60 | HECI2_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI2_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI2_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI2_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI2_H_GS2 | Host General Status 2 |
| 74 | HECI2_H_GS3 | Host General Status 3 |
| 8C | HECI2_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI2_MC | Message Signaled Interrupt Message Control |
| 90 | HECI2_MA | Message Signaled Interrupt Message Address |
| 94 | HECI2_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI2_MD | Message Signaled Interrupt Message Data |
| A0 | HECI2_HIDM | HECI Interrupt Delivery Mode |
| F8 | HECI2_MANID | Manufacturer's ID |

### 36.3.2.2 Memory-Mapped Registers

**Table 36-5. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI2_MMIO_MBAR (HI,LO) | 0 | HECI2_H_CB_WW | Host CB Write Window |
| HECI2_MMIO_MBAR (HI,LO) | 4 | HECI2_H_CSR | Host Control and Status Register |
| HECI2_MMIO_MBAR (HI,LO) | 8 | HECI2_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI2_MMIO_MBAR (HI,LO) | C | HECI2_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI2_MMIO_MBAR (HI,LO) | 800 | HECI2_D0I3C | D0i3 Control |

### 36.3.2.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 36.3.2.4 Sideband Registers

This device has no Sideband Registers.

## 36.3.3 Host Root Space - Intel® ME IDER

### 36.3.3.1 PCI Configuration and Capabilities

**Table 36-6. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D24:F2 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | IDE_HOST_DID_VID | Device ID and Vendor ID |
| 4 | IDE_HOST_STS_CMD | Status and Command |
| 8 | IDE_HOST_CC_RID | Class Code and Revision ID |
| C | IDE_HOST_BIST_HTYPE_LT_CLS | BIST, Header Type, Latency Timer, and Cache Line Size |
| 10 | IDE_HOST_PCMDIOBAR | IDE Primary Command Block IO BAR |
| 14 | IDE_HOST_PCTLIOBAR | IDE Primary Control Block IO BAR |
| 18 | IDE_HOST_SCMDIOBAR | IDE Secondary Command Block IO BAR |
| 1C | IDE_HOST_SCTLIOBAR | IDE Secondary Control Block IO BAR |
| 20 | IDE_HOST_BMIOBAR | IDE Bus Master Block IO BAR |
| 28 | IDE_HOST_CCP | Cardbus CIS Pointer |
| 2C | IDE_HOST_SID_SVID | Subsystem ID and Subsystem Vendor ID |
| 30 | IDE_HOST_XRBAR | Expansion ROM Base Address |
| 34 | IDE_HOST_CAPP | Capabilities List Pointer |
| 3C | IDE_HOST_MAXL_MING_INTP_INTL | Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line |
| 40 | IDE_HOST_MSIMC_MSINP_MSICID | MSI Message Control, Next Pointer and Capability ID |
| 44 | IDE_HOST_MSIMA | MSI Message Address |
| 48 | IDE_HOST_MSIMUA | MSI Message Upper Address |
| 4C | IDE_HOST_MSIMD | MSI Message Data |
| 50 | IDE_HOST_PMCAP_PMNP_PMCID | Power Management Capabilities, Next Pointer and Capability ID |
| 54 | IDE_HOST_PMD_PMCSRBSE_PMCSR | Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status |
| F8 | IDE_HOST_PSI | Process Stepping Information |

### 36.3.3.2 Memory-Mapped Registers

This device has no Memory-Mapped registers.

### 36.3.3.3 I/O-Mapped Registers

**Table 36-7. Relocatable I/O-Mapped Registers (Sheet 1 of 2)**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name | Note |
|---|---|---|---|---|
| IDE_HOST_PCMDIOBAR | 0 | IDEDATA | IDE Data Register | |
| IDE_HOST_PCMDIOBAR | 1 | IDEFR | IDE Features Register | Write |
| IDE_HOST_PCMDIOBAR | 1 | IDEERD0 | IDE Error Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 1 | IDEERD1 | IDE Error Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 2 | IDESCIR | IDE Sector Count In Register | Write |
| IDE_HOST_PCMDIOBAR | 2 | IDESCOR0 | IDE Sector Count Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 2 | IDESCOR1 | IDE Sector Count Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 3 | IDESNIR | IDE Sector Number In Register | Write |
| IDE_HOST_PCMDIOBAR | 3 | IDESNOR0 | IDE Sector Number Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 3 | IDESNOR1 | IDE Sector Number Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 4 | IDECLIR | IDE Cylinder Low In Register | Write |
| IDE_HOST_PCMDIOBAR | 4 | IDCLOR0 | IDE Cylinder Low Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 4 | IDCLOR1 | IDE Cylinder Low Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 5 | IDECHIR | IDE Cylinder High In Register | Write |
| IDE_HOST_PCMDIOBAR | 5 | IDCHOR0 | IDE Cylinder High Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 5 | IDCHOR1 | IDE Cylinder High Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 6 | IDEDHIR | IDE Drive/Head In Register | Write |
| IDE_HOST_PCMDIOBAR | 6 | IDDHOR0 | IDE Drive/Head Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 6 | IDDHOR1 | IDE Drive/Head Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 7 | IDECR | IDE Command Register | Write |
| IDE_HOST_PCMDIOBAR | 7 | IDESR0 | IDE Status Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 7 | IDESR1 | IDE Status Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCTLIOBAR | 2 | IDDCR | IDE Device Control Register | Write |
| IDE_HOST_PCTLIOBAR | 2 | IDASR | IDE Alternate Status Register | Read |
| IDE_HOST_SCMDIOBAR | - | - | Not in SRDL files | |
| IDE_HOST_SCTLIOBAR | - | - | Not in SRDL files | |
| IDE_HOST_BMIOBAR | 0 | IDEPBMCR | IDE Primary Bus Master Command Register | |
| IDE_HOST_BMIOBAR | 1 | IDEPBMDS0R | IDE Primary Bus Master Device Specific 0 Register | |
| IDE_HOST_BMIOBAR | 2 | IDEPBMSR | IDE Primary Bus Master Status Register | |
| IDE_HOST_BMIOBAR | 3 | IDEPBMDS1R | IDE Primary Bus Master Device Specific 1 Register | |
| IDE_HOST_BMIOBAR | 4 | IDEPBMDTPR0 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 | |
| IDE_HOST_BMIOBAR | 5 | IDEPBMDTPR1 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 | |
| IDE_HOST_BMIOBAR | 6 | IDEPBMDTPR2 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 | |
| IDE_HOST_BMIOBAR | 7 | IDEPBMDTPR3 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 | |

**Table 36-7. Relocatable I/O-Mapped Registers (Sheet 2 of 2)**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name | Note |
|---|---|---|---|---|
| IDE_HOST_BMIOBAR | 8 | IDESBMCR | IDE Secondary Bus Master Command Register | |
| IDE_HOST_BMIOBAR | 9 | IDESBMDS0R | IDE Secondary Bus Master Device Specific 0 Register | |
| IDE_HOST_BMIOBAR | A | IDESBMSR | IDE Secondary Bus Master Status Register | |
| IDE_HOST_BMIOBAR | B | IDESBMDS1R | IDE Secondary Bus Master Device Specific 1 Register | |
| IDE_HOST_BMIOBAR | C | IDESBMDTPR0 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 | |
| IDE_HOST_BMIOBAR | D | IDESBMDTPR1 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 | |
| IDE_HOST_BMIOBAR | E | IDESBMDTPR2 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 | |
| IDE_HOST_BMIOBAR | F | IDESBMDTPR3 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 | |

### 36.3.3.4 Sideband Registers

This device has no Sideband Registers.

## 36.3.4 Host Root Space - Intel® ME HECI3

### 36.3.4.1 PCI Configuration and Capabilities

**Table 36-8. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D24:F4 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI3_ID | Identifiers |
| 4 | HECI3_CMD | Command |
| 6 | HECI3_STS | Status |
| 8 | HECI3_RID_CC | Revision ID and Class Code |
| C | HECI3_CLS | Cache Line Size |
| D | HECI3_MLT | Master Latency Timer |
| E | HECI3_HTYPE | Header Type |
| F | HECI3_BIST | Built In Self-Test |
| 10 | HECI3_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI3_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI3_SS | Sub System Identifiers |
| 34 | HECI3_CAP | Capabilities Pointer |
| 3C | HECI3_INTR | Interrupt Information |
| 3E | HECI3_MGNT | Minimum Grant |
| 3F | HECI3_MLAT | Maximum Latency |
| 40 | HECI3_HFS | Host Firmware Status |
| 48 | HECI3_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI3_H_GS1 | Host General Status |
| 50 | HECI3_PID | PCI Power Management Capability ID |
| 52 | HECI3_PC | PCI Power Management Capabilities |
| 54 | HECI3_PMCS | PCI Power Management Control and Status |
| 60 | HECI3_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI3_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI3_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI3_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI3_H_GS2 | Host General Status 2 |
| 74 | HECI3_H_GS3 | Host General Status 3 |
| 8C | HECI3_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI3_MC | Message Signaled Interrupt Message Control |
| 90 | HECI3_MA | Message Signaled Interrupt Message Address |
| 94 | HECI3_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI3_MD | Message Signaled Interrupt Message Data |
| A0 | HECI3_HIDM | HECI Interrupt Delivery Mode |
| F8 | HECI3_MANID | Manufacturer's ID |

### 36.3.4.2 Memory-Mapped Registers

**Table 36-9. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI3_MMIO_MBAR (HI,LO) | 0 | HECI3_H_CB_WW | Host CB Write Window |
| HECI3_MMIO_MBAR (HI,LO) | 4 | HECI3_H_CSR | Host Control and Status Register |
| HECI3_MMIO_MBAR (HI,LO) | 8 | HECI3_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI3_MMIO_MBAR (HI,LO) | C | HECI3_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI3_MMIO_MBAR (HI,LO) | 800 | HECI3_D0I3C | D0i3 Control |

### 36.3.4.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 36.3.4.4 Sideband Registers

This device has no Sideband Registers.

## 36.3.5 Host Root Space - Intel® ME KT

### 36.3.5.1 PCI Configuration and Capabilities

**Table 36-10. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D24:F3 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | KT_HOST_DID_VID | Device ID and Vendor ID |
| 4 | KT_HOST_STS_CMD | Status and Command |
| 8 | KT_HOST_CC_RID | Class Code and Revision ID |
| C | KT_HOST_BIST_HTYPE_LT_CLS | BIST, Header Type, Latency Timer, and Cache Line Size |
| 10 | KT_HOST_IOBAR | KT IO BAR |
| 14 | KT_HOST_MEMBAR | KT Memory BAR |
| 28 | KT_HOST_CCP | Cardbus CIS Pointer |
| 2C | KT_HOST_SID_SVID | Subsystem ID and Subsystem Vendor ID |
| 30 | KT_HOST_XRBAR | Expansion ROM Base Address |
| 34 | KT_HOST_CAPP | Capabilities List Pointer |
| 3C | KT_HOST_MAXL_MING_INTP_INTL | Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line |
| 40 | KT_HOST_MSIMC_MSINP_MSICID | MSI Message Control, Next Pointer and Capability ID |
| 44 | KT_HOST_MSIMA | MSI Message Address |
| 48 | KT_HOST_MSIMUA | MSI Message Upper Address |
| 4C | KT_HOST_MSIMD | MSI Message Data |
| 50 | KT_HOST_PMCAP_PMNP_PMCID | Power Management Capabilities, Next Pointer and Capability ID |
| 54 | KT_HOST_PMD_PMCSRBSE_PMCSR | Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status |
| F8 | KT_HOST_PSI | Process Stepping Information |

### 36.3.5.2 Memory-Mapped Registers

**Table 36-11. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| KT_HOST_MEMBAR | 0 | MEM_KTRBR | KT Receive Buffer Register |
| KT_HOST_MEMBAR | 0 | MEM_KTTHR | KT Transmitter Holding Register |
| KT_HOST_MEMBAR | 0 | MEM_KTDLLR | KT Divisor Latch LSB Register |
| KT_HOST_MEMBAR | 1 | MEM_KTIER | KT Interrupt Enable register |
| KT_HOST_MEMBAR | 1 | MEM_KTDLMR | KT Divisor Latch MSB Register |
| KT_HOST_MEMBAR | 2 | MEM_KTIIR | KT Interrupt Identification Register |
| KT_HOST_MEMBAR | 2 | MEM_KTFCR | KT FIFO Control register |
| KT_HOST_MEMBAR | 3 | MEM_KTLCR | KT Line Control register |
| KT_HOST_MEMBAR | 4 | MEM_KTMCR | KT Modem Control register |
| KT_HOST_MEMBAR | 5 | MEM_KTLSR | KT Line Status register |
| KT_HOST_MEMBAR | 6 | MEM_KTMSR | KT Modem Status register |
| KT_HOST_MEMBAR | 7 | MEM_KTSCR | KT Scratch register |

### 36.3.5.3 I/O-Mapped Registers

**Table 36-12. Relocatable I/O-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| KT_HOST_IOBAR | 0 | IO_KTRBR | KT Receive Buffer Register |
| KT_HOST_IOBAR | 0 | IO_KTTHR | KT Transmitter Holding Register |
| KT_HOST_IOBAR | 0 | IO_KTDLLR | KT Divisor Latch LSB Register |
| KT_HOST_IOBAR | 1 | IO_KTIER | KT Interrupt Enable register |
| KT_HOST_IOBAR | 1 | IO_KTDLMR | KT Divisor Latch MSB Register |
| KT_HOST_IOBAR | 2 | IO_KTIIR | KT Interrupt Identification Register |
| KT_HOST_IOBAR | 2 | IO_KTFCR | KT FIFO Control register |
| KT_HOST_IOBAR | 3 | IO_KTLCR | KT Line Control register |
| KT_HOST_IOBAR | 4 | IO_KTMCR | KT Modem Control register |
| KT_HOST_IOBAR | 5 | IO_KTLSR | KT Line Status register |
| KT_HOST_IOBAR | 6 | IO_KTMSR | KT Modem Status register |
| KT_HOST_IOBAR | 7 | IO_KTSCR | KT Scratch register |

### 36.3.5.4 Sideband Registers

This device has no Sideband Registers.

§ §

# 37 Innovation Engine

The Innovation Engine (IE) is an optional, complete, embedded engine intended to enable SoC customers to provide their own custom system management. This chapter provides a brief overview of the IE. It is reserved for system-builder code, not for Intel firmware since Intel supplies IE hardware only. IE activation is not required for normal system operation.

*Note:* Intel recommends that customers perform the IE manufacturing lock step described, even if not using the Innovation Engine, to eliminate any in field security risk.

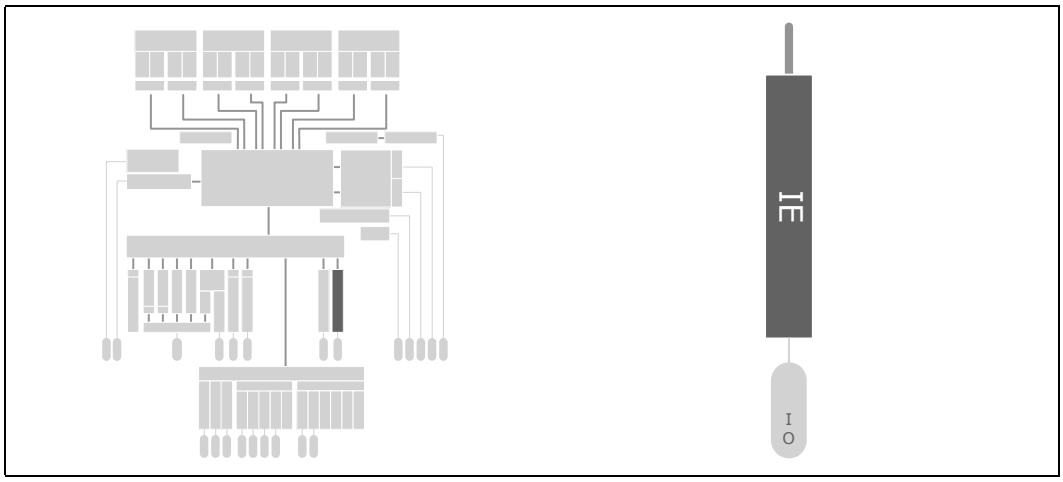
IE is a completely optional feature, and is disabled by default in the silicon. It can be enabled by system builders and OEMs to run signed firmware created by the system builder or a third party software vendor. IE is not like the Intel® Management Engine (Intel® ME) where Intel provides the HW plus a complete FW solution. Intel only provides IE hardware (along with collateral and tools enabling). If the IE is configured to be enabled, the IE ROM brings up the system in real mode and then hands off to the system builder's firmware to continue.

To use IE, the OEM enables the IE in Soft Straps, in the SPI descriptor region (using the Flash Image Tool).

Customers should be resourced and able to develop their own firmware for the IE, or work with a third party vendor to develop IE firmware solutions.

*Note:* If the OEM has enabled IE Verified Boot, then the IE will always be started, and the IE Soft Strap Enable/Disable is ignored.

**Figure 37-1. What is Covered in This Chapter**



The signal descriptions are shown in Table 37-1, "Signal Names and Descriptions." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Section 39.1, "Directory of Signal Names and Pin Names."

The Direction/Type column of Table 37-1, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

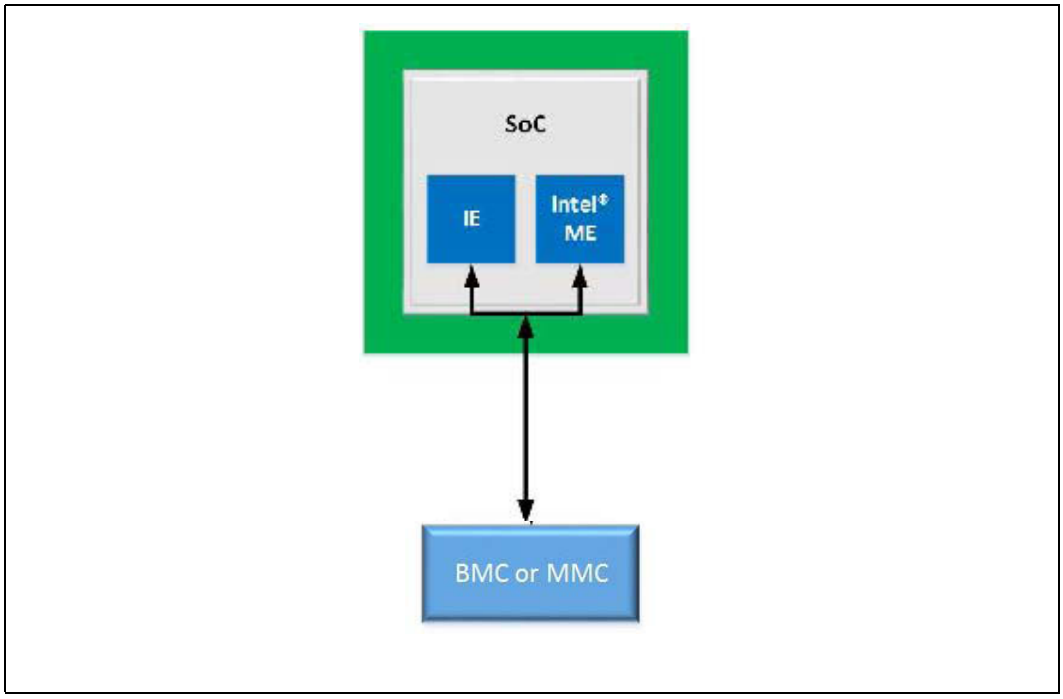
## Table 37-1. Signal Names and Descriptions

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| UART_IE_RXD | I | | UART_IE Receive Data: Active-high input. See Table 18-2, "Signal Names and Descriptions" on page 566. |
| UART_IE_TXD | O | | UART_IE Transmit Data: Active-high output. See Table 18-2, "Signal Names and Descriptions" on page 566. |
| UART_IE_CTS | I | | UART_IE Clear to Send: Active-high input. See Table 18-2, "Signal Names and Descriptions" on page 566. |
| UART_IE_RTS | O | | UART_IE Request to Send: Active-high output. See Table 18-2, "Signal Names and Descriptions" on page 566. |
| SMB_IE_SMT0_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 0 of the IE. |
| SMB_IE_SMT0_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 0 of the IE. |
| SMB_IE_SMT0_ALRT_N | I,O-OD | Yes | **SMBus Alert:** (SMBALERT#) for System Management Transport 0 of the IE. |
| SMB_IE_SMT1_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 1 of the IE. |
| SMB_IE_SMT1_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 1 of the IE. |
| SMB_IE_SMT2_CLK | I,O-OD | Yes | **SMBus Clock:** (SMBCLK) for System Management Transport 2 of the IE. |
| SMB_IE_SMT2_DATA | I,O-OD | Yes | **SMBus Data:** (SMBDAT) for System Management Transport 2 of the IE. |
| FAN_PWM[3:0] | O-OD | Yes | **Fan Pulse-Width Modulation (PWM):** These open-drain output signals produce a PWM duty cycle that is used for fan-speed control. When controlling a 3-wire-controlled fan, this signal controls a power transistor that, in turn, controls power to the fan. When controlling a 4-wire-controlled fan, this signal is connected to the "Control" signal on the fan. The polarity of this signal is programmable. The fan controller is part of the IE. |
| FAN_TACH[3:0] | I | Yes | **Fan Tachometer:** This pulsed input signal is used by the IE to measure fan speed. This signal is connected to the "Sense" signal on the fan. This signal is driven by the fan in an open-drain configuration. |

## 37.1 Feature Summary

- IE is an embedded sub-system in the SoC:
  - Includes a small, 32-bit IA core.
- Reserved for system-builder's code, not for Intel firmware:
  - Intel supplies IE hardware only.
  - IE code is cryptographically signed by the system builder using a private key.
  - IE code that doesn't authenticate with the system builder public key will not load.
- IE activation is not required for normal system operation:
  - Completely optional feature.
- Power:
  - When in S0, the SoC can be configured by BIOS to completely shut down the IE power.
  - The SoC has a power-disable policy that can be set to shut down the IE and all its contents are lost.

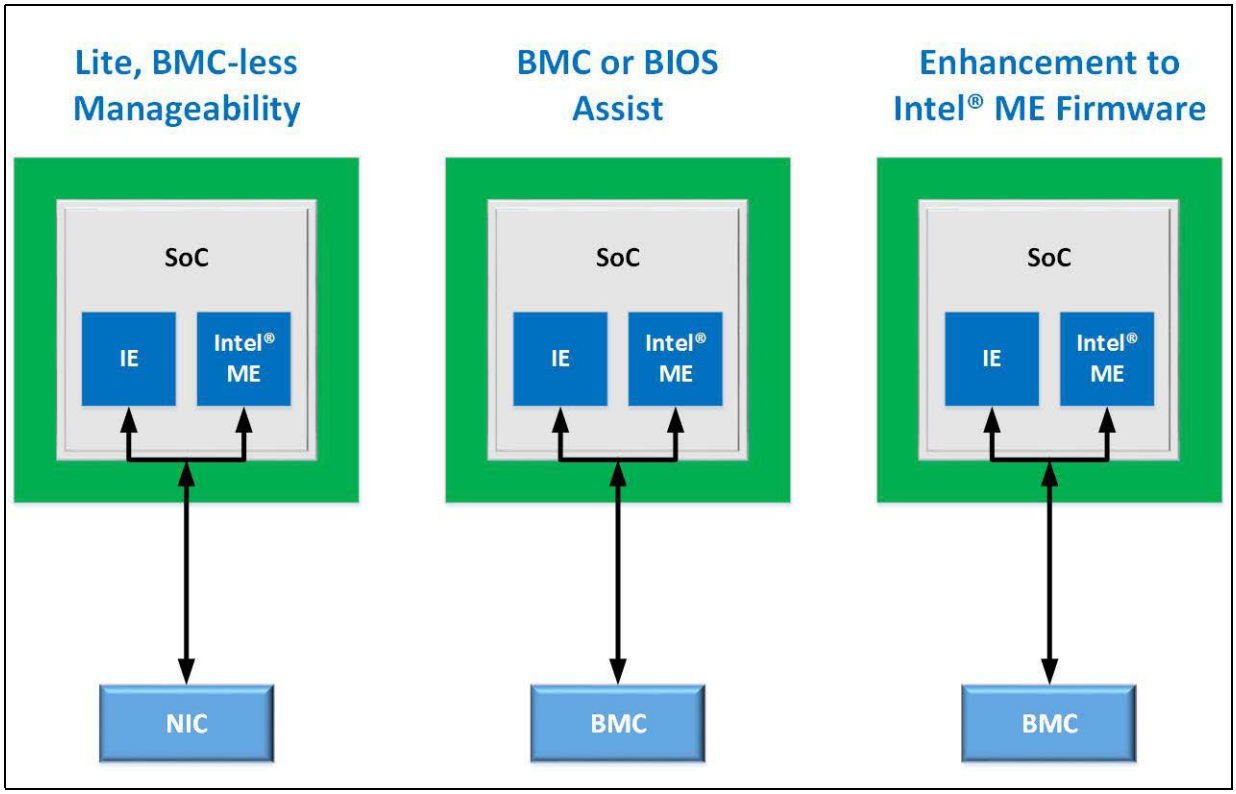
**Figure 37-2. Intel Atom® Processor C3000 Product Family Embedded Sub-System**



## 37.2 Sample IE Usages

- Lightweight Management/BMC-less platforms. IE takes role of BMC and can do lightweight BMC functions including log SEL events, serial over LAN, basic sensor monitoring. Can also do an IPMI 2.0 compliant implementation.
- BMC or BIOS assist – Enterprise segment likes video redirection and higher BMC capabilities. BMC can respond to network packets and present a nice GUI over network, offload tight loop code to IE
- Enhancements to Intel® ME FW – to optimize or customize for specific usages.
- IE Verified Boot – IE code signed by the system builder is authenticated. The system builder code then authenticates the BIOS image.

**Figure 37-3. Sample IE Usages**



## 37.3 Architectural Overview

The Innovation Engine (IE) is an integrated SoC subsystem that is based on a custom Intel® Architecture Processor. This subsystem has dedicated general purpose DMA and Crypto accelerator engines. The PCI Express Message Transport subsystem is comprised of I/O devices, based on standard PCI architecture. This subsystem is connected to the rest of the system via internal fabric, if required. This subsystem maintains its own system space and exposes some of its interface to the Host space, as described below.

The IE ROM holds boot code and is responsible for initialization of the IE subsystem. The subsystem authenticates the OEM IE image and initiates the start up process.  SPI Flash is used to store OEM IE images.

IE Memory Subsystem – (SRAM and ROM) – completely independent from Host and Intel® ME.

- IE SRAM – 1.4MB available for IE to run FW.
- IE ROM – 128KB of dedicated ROM embedded in silicon (hard-coded, read-only).

The shared SPI Flash – dedicated IE region for customer FW storage (4-8MB). Regions are defined for all masters accessing the shared SPI flash. Those regions can be as big as the overall SPI flash size (so full flexibility). But in most cases, would be a 4-8MB range allocated for IE in typical platforms.

Depending on what application is running, SPI flash could be used for storing additional data, or holding a small image that could be exposed through IE. We also support paging from SPI flash. Code and data can be bigger than on-die SRAM and can page out of SPI flash. This paging would be managed as a kernel function (have to support page fault handling in kernel), plus DMA driver to load a 4K page from SPI flash into SRAM.

The devices exposed by the IE subsystem to the Host Root Space are:

- HECI (1, 2 and 3) – These functions define the mechanism for host software and IE firmware to communicate. This device exposes three PCI functions to the host during PCI bus enumeration. The message format is OEM dependent and communication between host and IE subsystem takes place via circular buffers and control/status registers. This function supports host MSI, SMI and SCI# interrupt generation mechanisms.
- PTIO – This exposes two PCI functions to the host.
  - Keyboard Text (KT) – The Innovation Engine provides a Keyboard Transfer (KT) interface to redirect virtual keyboard input to the host. The interface is visible to the Innovation Engine as a PCI device.
  - IDE-Redirect (IDER) - The IE provides an IDE-redirection mechanism that performs the basic functions. It provides the capability to boot from a remote IDE drive.

Some of Host Root Space devices are also part of IE root space. A partial list of devices parts of the IE root space are:

- HECI devices/clients
  - HECI specification uses term “HECI client” to refer to software endpoints that send and receive messages via HECI.
  - Examples of Host HECI clients are System BIOS module, OS Services & Applications that interface with HECI.
  - FW HECI Clients are IE FW modules that use HECI to communicate with Host Clients.
  - Each endpoint supports many connections established by the clients at the same time as each client is identified by a unique address assigned while establishing the connection. Both host client address and FW client address are contained by the HECI message header to enable dispatch of the message. HECI Protocol doesn't enforce any restrictions on the format of the message data between HECI Clients.

*Note:* IE HSUART and IE SMT are exclusively IE Root Space devices. They are not discovered or addressable in Host Root Space.

- HSUART - UART primary fabric interface only supports single channel and 32-bit data/addressing. The register space is not mapped into the prefetchable address space, and the primary target interface does not support more than one DWord access per I/O fabric request.

  In the downstream direction, a register access of more than one DWord could result in undefined behavior and will be treated as an unsupported request. Any unsupported downstream Posted request will be dropped and unsupported, Non-Posted request will be completed with the appropriate unsuccessful completion status. The first 2 DWs of each UART CSR are IO as well as memory mapped. The UART I/O fabric is a PCI device and supports configuration read and write requests at the primary target.

  In the upstream direction, the primary interface makes 32-bit memory read and write requests for DMA transfers. UART primary fabric also supports single MSI vector per function which will be sent as a Posted Write request in upstream direction.

  UART only supports 64 bytes I/O fabric Max Payload Size and Max Read Request Size.

  All upstream memory requests will be split at 64byte block boundary. As no read request will cross the 64bytes RCB, the UART I/O fabric accepts only single completions for each upstream memory read request.

- SMBus Message Transport (SMT): There are three SMBus devices in the IE root space. The SMT provides a mechanism to the IE for sending OOB manageability messages to managed devices or management controllers over SMBus.

## 37.4 Inter-Processor Communication (IPC)

The IE communicates with the Intel® Management Engine using an IPC Interface over the sideband fabric. The IPC Interface is intended for transferring small messages directly between cores. IPC messages from the IE core are routed to the sideband fabric.

### 37.4.1 IE Root Space Devices

In addition to the devices that are visible to both the Host address space and the IE root space, the following devices are only visible internally to the IE root space.

#### 37.4.1.1 PECI

In-band PECI is restricted to Intel® ME. PECI over SMBus is also supported through an external wire.

#### 37.4.1.2 SMBus Message Transport (SMT)

The SoC provides a number of virtual SMBus interfaces that are visible to the IE Root Space as PCI devices. Externally, these interfaces conform to the I2C specification as physical SMBus devices. The configuration of the SMT interfaces is determined at manufacturing time, and can be configured to the Intel® Management Engine (Intel® ME), or the Innovation Engine. The requirements vary depending on which Intel® ME features are desired by the system builder.

Bus speed can be configured by Soft Straps to run I2C Fast-mode Plus (Fm+) timings (up to 1MHz) if all devices on the segment support the higher speed.

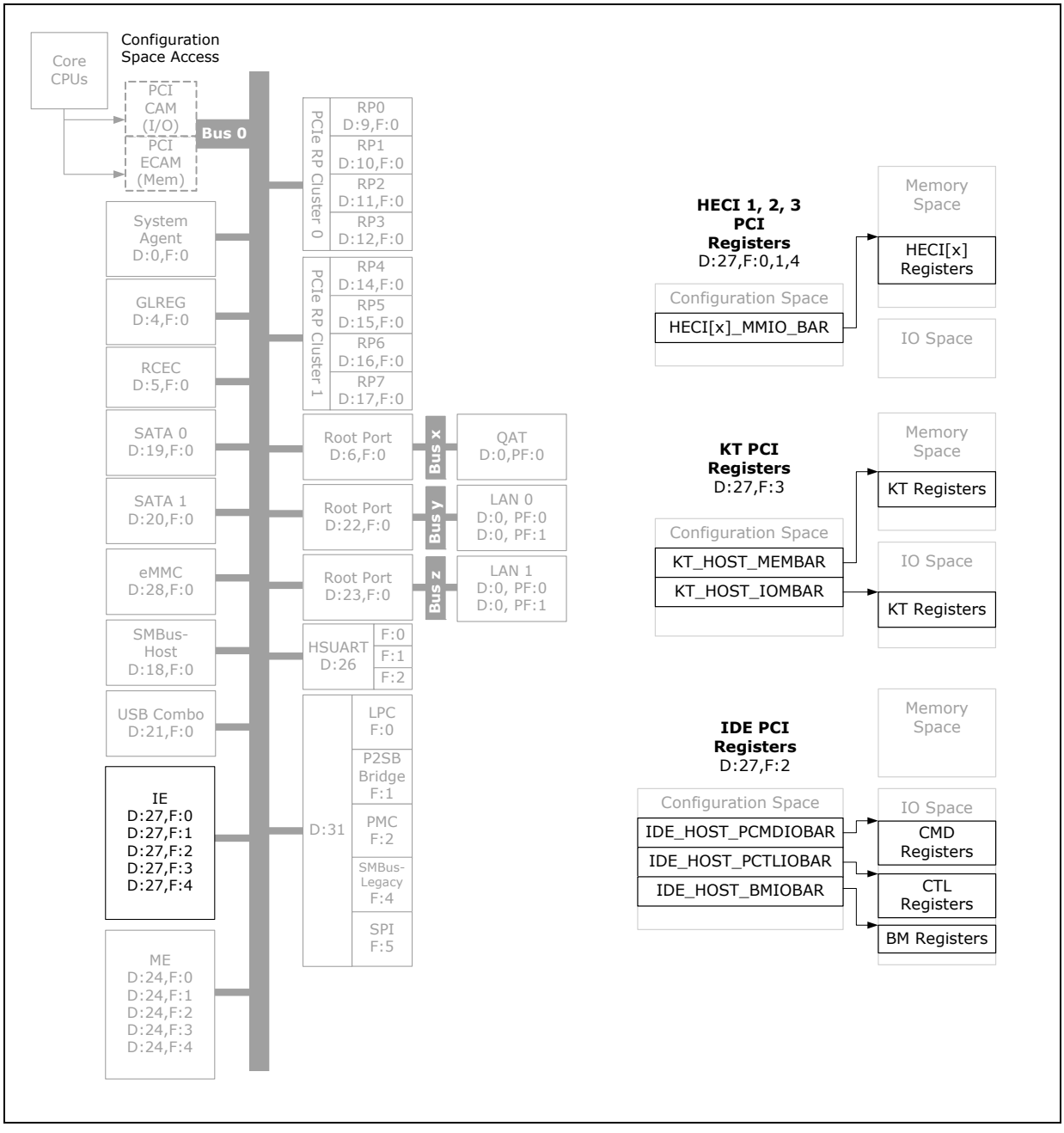
#### 37.4.1.3 UART

The Innovation Engine supports a serial UART for communication between the Innovation Engine and an external interface, such as an OEM Chassis Management Module. Additionally, this interface can be used for debug purposes.

## 37.5 Register Map

The IE is accessible through the Host Root Space as shown in Figure 37-4, "Register Map."

**Figure 37-4. Register Map**



## 37.5.1 Host Root Space - IE HECI1

### 37.5.1.1 PCI Configuration and Capabilities

**Table 37-2. Configuration and Capabilities Register Map (Sheet 1 of 2)**

| Offset from Configuration Space of B0:D27:F0 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI1_ID | Identifiers |
| 4 | HECI1_CMD | Command |
| 6 | HECI1_STS | Status |
| 8 | HECI1_RID_CC | Revision ID and Class Code |
| C | HECI1_CLS | Cache Line Size |
| D | HECI1_MLT | Master Latency Timer |
| E | HECI1_HTYPE | Header Type |
| F | HECI1_BIST | Built In Self-Test |
| 10 | HECI1_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI1_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI1_SS | Sub System Identifiers |
| 34 | HECI1_CAP | Capabilities Pointer |
| 3C | HECI1_INTR | Interrupt Information |
| 3E | HECI1_MGNT | Minimum Grant |
| 3F | HECI1_MLAT | Maximum Latency |
| 40 | HECI1_HFS | Host Firmware Status |
| 44 | HECI1_MISC_SHDW | Miscellaneous Shadow |
| 48 | HECI1_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI1_H_GS1 | Host General Status |
| 50 | HECI1_PID | PCI Power Management Capability ID |
| 52 | HECI1_PC | PCI Power Management Capabilities |
| 54 | HECI1_PMCS | PCI Power Management Control and Status |
| 60 | HECI1_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI1_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI1_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI1_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI1_H_GS2 | Host General Status 2 |
| 74 | HECI1_H_GS3 | Host General Status 3 |
| 8C | HECI1_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI1_MC | Message Signaled Interrupt Message Control |
| 90 | HECI1_MA | Message Signaled Interrupt Message Address |
| 94 | HECI1_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI1_MD | Message Signaled Interrupt Message Data |
| A0 | HECI1_HIDM | HECI Interrupt Delivery Mode |

## Table 37-2. Configuration and Capabilities Register Map (Sheet 2 of 2)

| Offset from Configuration Space of B0:D27:F0 (hex) | Register Short Name | Description |
|---|---|---|
| BC | HECI1_HERS | Host Extend Register Status |
| C0 | HECI1_HER1 | Host Extend Register DW1. |
| C4 | HECI1_HER2 | Host Extend Register DW2. |
| C8 | HECI1_HER3 | Host Extend Register DW3. |
| CC | HECI1_HER4 | Host Extend Register DW4. |
| D0 | HECI1_HER5 | Host Extend Register DW5. |
| D4 | HECI1_HER6 | Host Extend Register DW6. |
| D8 | HECI1_HER7 | Host Extend Register DW7. |
| DC | HECI1_HER8 | Host Extend Register DW8. |
| F8 | HECI1_MANID | Manufacturer's ID |

### 37.5.1.2 Memory-Mapped Registers

**Table 37-3. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI1_MMIO_MBAR (HI,LO) | 0 | HECI1_H_CB_WW | Host CB Write Window |
| HECI1_MMIO_MBAR (HI,LO) | 4 | HECI1_H_CSR | Host Control and Status Register |
| HECI1_MMIO_MBAR (HI,LO) | 8 | HECI1_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI1_MMIO_MBAR (HI,LO) | C | HECI1_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI1_MMIO_MBAR (HI,LO) | 800 | HECI1_D0I3C | D0i3 Control |

### 37.5.1.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 37.5.1.4 Sideband Registers

This device has no Sideband Registers.

## 37.5.2 Host Root Space - IE HECI2

### 37.5.2.1 PCI Configuration and Capabilities

**Table 37-4. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D27:F1 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI2_ID | Identifiers |
| 4 | HECI2_CMD | Command |
| 6 | HECI2_STS | Status |
| 8 | HECI2_RID_CC | Revision ID and Class Code |
| C | HECI2_CLS | Cache Line Size |
| D | HECI2_MLT | Master Latency Timer |
| E | HECI2_HTYPE | Header Type |
| F | HECI2_BIST | Built In Self-Test |
| 10 | HECI2_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI2_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI2_SS | Sub System Identifiers |
| 34 | HECI2_CAP | Capabilities Pointer |
| 3C | HECI2_INTR | Interrupt Information |
| 3E | HECI2_MGNT | Minimum Grant |
| 3F | HECI2_MLAT | Maximum Latency |
| 40 | HECI2_HFS | Host Firmware Status |
| 48 | HECI2_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI2_H_GS1 | Host General Status |
| 50 | HECI2_PID | PCI Power Management Capability ID |
| 52 | HECI2_PC | PCI Power Management Capabilities |
| 54 | HECI2_PMCS | PCI Power Management Control and Status |
| 60 | HECI2_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI2_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI2_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI2_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI2_H_GS2 | Host General Status 2 |
| 74 | HECI2_H_GS3 | Host General Status 3 |
| 8C | HECI2_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI2_MC | Message Signaled Interrupt Message Control |
| 90 | HECI2_MA | Message Signaled Interrupt Message Address |
| 94 | HECI2_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI2_MD | Message Signaled Interrupt Message Data |
| A0 | HECI2_HIDM | HECI Interrupt Delivery Mode |
| F8 | HECI2_MANID | Manufacturer's ID |

### 37.5.2.2 Memory-Mapped Registers

**Table 37-5. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI2_MMIO_MBAR (HI,LO) | 0 | HECI2_H_CB_WW | Host CB Write Window |
| HECI2_MMIO_MBAR (HI,LO) | 4 | HECI2_H_CSR | Host Control and Status Register |
| HECI2_MMIO_MBAR (HI,LO) | 8 | HECI2_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI2_MMIO_MBAR (HI,LO) | C | HECI2_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI2_MMIO_MBAR (HI,LO) | 800 | HECI2_D0I3C | D0i3 Control |

### 37.5.2.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 37.5.2.4 Sideband Registers

This device has no Sideband Registers.

## 37.5.3 Host Root Space - IE IDER

### 37.5.3.1 PCI Configuration and Capabilities

**Table 37-6. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D27:F2 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | IDE_HOST_DID_VID | Device ID and Vendor ID |
| 4 | IDE_HOST_STS_CMD | Status and Command |
| 8 | IDE_HOST_CC_RID | Class Code and Revision ID |
| C | IDE_HOST_BIST_HTYPE_LT_CLS | BIST, Header Type, Latency Timer, and Cache Line Size |
| 10 | IDE_HOST_PCMDIOBAR | IDE Primary Command Block IO BAR |
| 14 | IDE_HOST_PCTLIOBAR | IDE Primary Control Block IO BAR |
| 18 | IDE_HOST_SCMDIOBAR | IDE Secondary Command Block IO BAR |
| 1C | IDE_HOST_SCTLIOBAR | IDE Secondary Control Block IO BAR |
| 20 | IDE_HOST_BMIOBAR | IDE Bus Master Block IO BAR |
| 28 | IDE_HOST_CCP | Cardbus CIS Pointer |
| 2C | IDE_HOST_SID_SVID | Subsystem ID and Subsystem Vendor ID |
| 30 | IDE_HOST_XRBAR | Expansion ROM Base Address |
| 34 | IDE_HOST_CAPP | Capabilities List Pointer |
| 3C | IDE_HOST_MAXL_MING_INTP_INTL | Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line |
| 40 | IDE_HOST_MSIMC_MSINP_MSICID | MSI Message Control, Next Pointer and Capability ID |
| 44 | IDE_HOST_MSIMA | MSI Message Address |
| 48 | IDE_HOST_MSIMUA | MSI Message Upper Address |
| 4C | IDE_HOST_MSIMD | MSI Message Data |
| 50 | IDE_HOST_PMCAP_PMNP_PMCID | Power Management Capabilities, Next Pointer and Capability ID |
| 54 | IDE_HOST_PMD_PMCSRBSE_PMCSR | Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status |
| F8 | IDE_HOST_PSI | Process Stepping Information |

### 37.5.3.2 Memory-Mapped Registers

This device has no Memory-Mapped registers.

### 37.5.3.3 I/O-Mapped Registers

**Table 37-7. Relocatable I/O-Mapped Registers (Sheet 1 of 2)**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name | Note |
|---|---|---|---|---|
| IDE_HOST_PCMDIOBAR | 0 | IDEDATA | IDE Data Register | |
| IDE_HOST_PCMDIOBAR | 1 | IDEFR | IDE Features Register | Write |
| IDE_HOST_PCMDIOBAR | 1 | IDEERD0 | IDE Error Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 1 | IDEERD1 | IDE Error Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 2 | IDESCIR | IDE Sector Count In Register | Write |
| IDE_HOST_PCMDIOBAR | 2 | IDESCOR0 | IDE Sector Count Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 2 | IDESCOR1 | IDE Sector Count Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 3 | IDESNIR | IDE Sector Number In Register | Write |
| IDE_HOST_PCMDIOBAR | 3 | IDESNOR0 | IDE Sector Number Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 3 | IDESNOR1 | IDE Sector Number Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 4 | IDECLIR | IDE Cylinder Low In Register | Write |
| IDE_HOST_PCMDIOBAR | 4 | IDCLOR0 | IDE Cylinder Low Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 4 | IDCLOR1 | IDE Cylinder Low Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 5 | IDECHIR | IDE Cylinder High In Register | Write |
| IDE_HOST_PCMDIOBAR | 5 | IDCHOR0 | IDE Cylinder High Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 5 | IDCHOR1 | IDE Cylinder High Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 6 | IDEDHIR | IDE Drive/Head In Register | Write |
| IDE_HOST_PCMDIOBAR | 6 | IDDHOR0 | IDE Drive/Head Out Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 6 | IDDHOR1 | IDE Drive/Head Out Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCMDIOBAR | 7 | IDECR | IDE Command Register | Write |
| IDE_HOST_PCMDIOBAR | 7 | IDESR0 | IDE Status Register DEV0 | Read and Dev = 0 |
| IDE_HOST_PCMDIOBAR | 7 | IDESR1 | IDE Status Register DEV1 | Read and Dev = 1 |
| IDE_HOST_PCTLIOBAR | 2 | IDDCR | IDE Device Control Register | Write |
| IDE_HOST_PCTLIOBAR | 2 | IDASR | IDE Alternate Status Register | Read |
| IDE_HOST_BMIOBAR | 0 | IDEPBMCR | IDE Primary Bus Master Command Register | |
| IDE_HOST_BMIOBAR | 1 | IDEPBMDS0R | IDE Primary Bus Master Device Specific 0 Register | |
| IDE_HOST_BMIOBAR | 2 | IDEPBMSR | IDE Primary Bus Master Status Register | |
| IDE_HOST_BMIOBAR | 3 | IDEPBMDS1R | IDE Primary Bus Master Device Specific 1 Register | |
| IDE_HOST_BMIOBAR | 4 | IDEPBMDTPR0 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 | |
| IDE_HOST_BMIOBAR | 5 | IDEPBMDTPR1 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 | |
| IDE_HOST_BMIOBAR | 6 | IDEPBMDTPR2 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 | |

**Table 37-7. Relocatable I/O-Mapped Registers (Sheet 2 of 2)**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name | Note |
|---|---|---|---|---|
| IDE_HOST_BMIOBAR | 7 | IDEPBMDTPR3 | IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 | |
| IDE_HOST_BMIOBAR | 8 | IDESBMCR | IDE Secondary Bus Master Command Register | |
| IDE_HOST_BMIOBAR | 9 | IDESBMDS0R | IDE Secondary Bus Master Device Specific 0 Register | |
| IDE_HOST_BMIOBAR | A | IDESBMSR | IDE Secondary Bus Master Status Register | |
| IDE_HOST_BMIOBAR | B | IDESBMDS1R | IDE Secondary Bus Master Device Specific 1 Register | |
| IDE_HOST_BMIOBAR | C | IDESBMDTPR0 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 | |
| IDE_HOST_BMIOBAR | D | IDESBMDTPR1 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 | |
| IDE_HOST_BMIOBAR | E | IDESBMDTPR2 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 | |
| IDE_HOST_BMIOBAR | F | IDESBMDTPR3 | IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 | |

### 37.5.3.4 Sideband Registers

This device has no Sideband Registers.

## 37.5.4 Host Root Space - IE HECI3

### 37.5.4.1 PCI Configuration and Capabilities

**Table 37-8. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D27:F4 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | HECI3_ID | Identifiers |
| 4 | HECI3_CMD | Command |
| 6 | HECI3_STS | Status |
| 8 | HECI3_RID_CC | Revision ID and Class Code |
| C | HECI3_CLS | Cache Line Size |
| D | HECI3_MLT | Master Latency Timer |
| E | HECI3_HTYPE | Header Type |
| F | HECI3_BIST | Built In Self-Test |
| 10 | HECI3_MMIO_MBAR_LO | HECI MMIO Base Address Low |
| 14 | HECI3_MMIO_MBAR_HI | HECI MMIO Base Address High |
| 2C | HECI3_SS | Sub System Identifiers |
| 34 | HECI3_CAP | Capabilities Pointer |
| 3C | HECI3_INTR | Interrupt Information |
| 3E | HECI3_MGNT | Minimum Grant |
| 3F | HECI3_MLAT | Maximum Latency |
| 40 | HECI3_HFS | Host Firmware Status |
| 48 | HECI3_GS_SHDW1 | General Status Shadow 1 |
| 4C | HECI3_H_GS1 | Host General Status |
| 50 | HECI3_PID | PCI Power Management Capability ID |
| 52 | HECI3_PC | PCI Power Management Capabilities |
| 54 | HECI3_PMCS | PCI Power Management Control and Status |
| 60 | HECI3_GS_SHDW2 | General Status Shadow 2 |
| 64 | HECI3_GS_SHDW3 | General Status Shadow 3 |
| 68 | HECI3_GS_SHDW4 | General Status Shadow 4 |
| 6C | HECI3_GS_SHDW5 | General Status Shadow 5 |
| 70 | HECI3_H_GS2 | Host General Status 2 |
| 74 | HECI3_H_GS3 | Host General Status 3 |
| 8C | HECI3_MID | Message Signaled Interrupt Identifiers |
| 8E | HECI3_MC | Message Signaled Interrupt Message Control |
| 90 | HECI3_MA | Message Signaled Interrupt Message Address |
| 94 | HECI3_MUA | Message Signaled Interrupt Upper Address |
| 98 | HECI3_MD | Message Signaled Interrupt Message Data |
| A0 | HECI3_HIDM | HECI Interrupt Delivery Mode |
| F8 | HECI3_MANID | Manufacturer's ID |

### 37.5.4.2 Memory-Mapped Registers

**Table 37-9. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| HECI3_MMIO_MBAR (HI,LO) | 0 | HECI3_H_CB_WW | Host CB Write Window |
| HECI3_MMIO_MBAR (HI,LO) | 4 | HECI3_H_CSR | Host Control and Status Register |
| HECI3_MMIO_MBAR (HI,LO) | 8 | HECI3_CSE_CB_RW | CSE Circular Buffer Read Window |
| HECI3_MMIO_MBAR (HI,LO) | C | HECI3_CSE_CSR_HA | CSE Control and Status Register Host Access |
| HECI3_MMIO_MBAR (HI,LO) | 800 | HECI3_D0I3C | D0i3 Control |

### 37.5.4.3 I/O-Mapped Registers

This device has no I/O-mapped registers.

### 37.5.4.4 Sideband Registers

This device has no Sideband Registers.

## 37.5.5 Host Root Space - IE KT

### 37.5.5.1 PCI Configuration and Capabilities

**Table 37-10. Configuration and Capabilities Register Map**

| Offset from Configuration Space of B0:D27:F3 (hex) | Register Short Name | Description |
|---|---|---|
| 0 | KT_HOST_DID_VID | Device ID and Vendor ID |
| 4 | KT_HOST_STS_CMD | Status and Command |
| 8 | KT_HOST_CC_RID | Class Code and Revision ID |
| C | KT_HOST_BIST_HTYPE_LT_CLS | BIST, Header Type, Latency Timer, and Cache Line Size |
| 10 | KT_HOST_IOBAR | KT IO BAR |
| 14 | KT_HOST_MEMBAR | KT Memory BAR |
| 28 | KT_HOST_CCP | Cardbus CIS Pointer |
| 2C | KT_HOST_SID_SVID | Subsystem ID and Subsystem Vendor ID |
| 30 | KT_HOST_XRBAR | Expansion ROM Base Address |
| 34 | KT_HOST_CAPP | Capabilities List Pointer |
| 3C | KT_HOST_MAXL_MING_INTP_INTL | Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line |
| 40 | KT_HOST_MSIMC_MSINP_MSICID | MSI Message Control, Next Pointer and Capability ID |
| 44 | KT_HOST_MSIMA | MSI Message Address |
| 48 | KT_HOST_MSIMUA | MSI Message Upper Address |
| 4C | KT_HOST_MSIMD | MSI Message Data |
| 50 | KT_HOST_PMCAP_PMNP_PMCID | Power Management Capabilities, Next Pointer and Capability ID |
| 54 | KT_HOST_PMD_PMCSRBSE_PMCSR | Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status |
| F8 | KT_HOST_PSI | Process Stepping Information |

### 37.5.5.2 Memory-Mapped Registers

**Table 37-11. Relocatable Memory-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| KT_HOST_MEMBAR | 0 | MEM_KTRBR | KT Receive Buffer Register |
| KT_HOST_MEMBAR | 0 | MEM_KTTHR | KT Transmitter Holding Register |
| KT_HOST_MEMBAR | 0 | MEM_KTDLLR | KT Divisor Latch LSB Register |
| KT_HOST_MEMBAR | 1 | MEM_KTIER | KT Interrupt Enable register |
| KT_HOST_MEMBAR | 1 | MEM_KTDLMR | KT Divisor Latch MSB Register |
| KT_HOST_MEMBAR | 2 | MEM_KTIIR | KT Interrupt Identification Register |
| KT_HOST_MEMBAR | 2 | MEM_KTFCR | KT FIFO Control register |
| KT_HOST_MEMBAR | 3 | MEM_KTLCR | KT Line Control register |
| KT_HOST_MEMBAR | 4 | MEM_KTMCR | KT Modem Control register |
| KT_HOST_MEMBAR | 5 | MEM_KTLSR | KT Line Status register |
| KT_HOST_MEMBAR | 6 | MEM_KTMSR | KT Modem Status register |
| KT_HOST_MEMBAR | 7 | MEM_KTSCR | KT Scratch register |

### 37.5.5.3 I/O-Mapped Registers

**Table 37-12. Relocatable I/O-Mapped Registers**

| Base Address Register (BAR) Short Name | Offset from BAR (hex) | Register Short Name | Register Long Name |
|---|---|---|---|
| KT_HOST_IOBAR | 0 | IO_KTRBR | KT Receive Buffer Register |
| KT_HOST_IOBAR | 0 | IO_KTTHR | KT Transmitter Holding Register |
| KT_HOST_IOBAR | 0 | IO_KTDLLR | KT Divisor Latch LSB Register |
| KT_HOST_IOBAR | 1 | IO_KTIER | KT Interrupt Enable register |
| KT_HOST_IOBAR | 1 | IO_KTDLMR | KT Divisor Latch MSB Register |
| KT_HOST_IOBAR | 2 | IO_KTIIR | KT Interrupt Identification Register |
| KT_HOST_IOBAR | 2 | IO_KTFCR | KT FIFO Control register |
| KT_HOST_IOBAR | 3 | IO_KTLCR | KT Line Control register |
| KT_HOST_IOBAR | 4 | IO_KTMCR | KT Modem Control register |
| KT_HOST_IOBAR | 5 | IO_KTLSR | KT Line Status register |
| KT_HOST_IOBAR | 6 | IO_KTMSR | KT Modem Status register |
| KT_HOST_IOBAR | 7 | IO_KTSCR | KT Scratch register |

### 37.5.5.4 Sideband Registers

This device has no Sideband Registers.

§ §

# 38 JTAG and Debug Ports

This chapter describes the SoC debug mechanisms available to customers. This includes the Intel® Direct Connect Interface (Intel® DCI) and Joint Test Action Group (JTAG) interface.

**Figure 38-1. What is Covered in This Chapter**

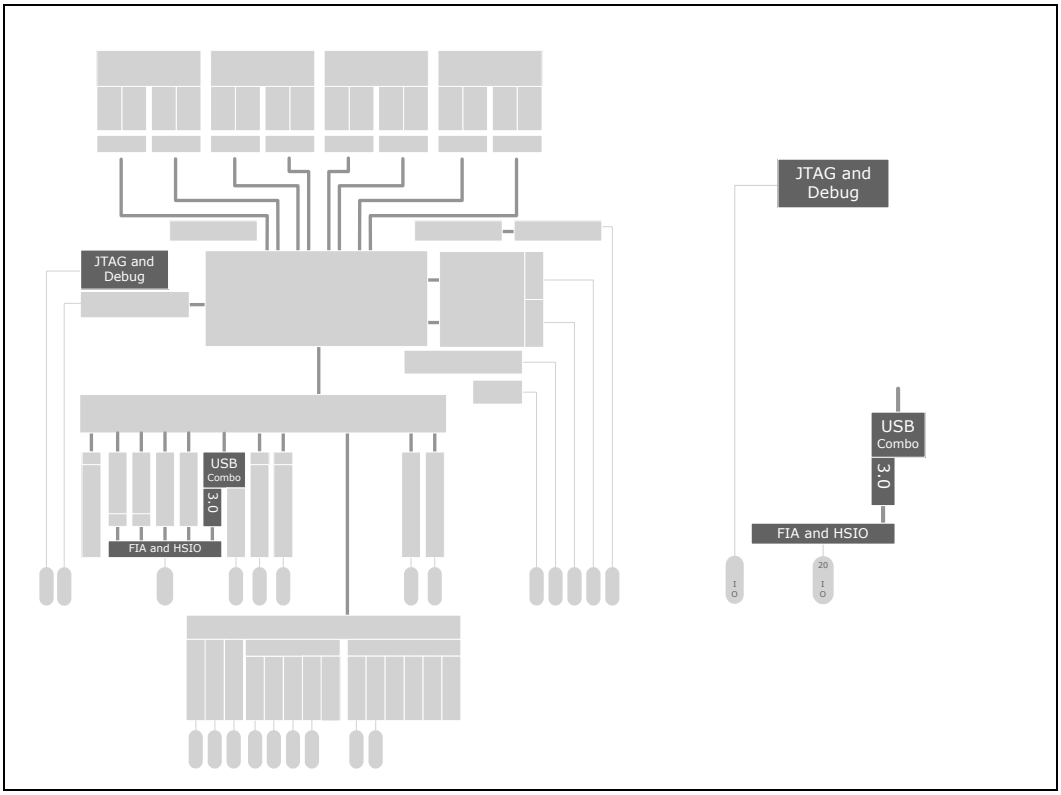



**Table 38-1. References**

| Reference | Revision | Date | Document Title |
|---|---|---|---|
| IEEE Std 1149.1*-2001 | - | June 14, 2001 | *IEEE Standard Test Access Port and Boundary-Scan Architecture*, 1149.1-2001 |
| IEEE Std 1149.7*-2009 | - | Dec. 9, 2009 | *IEEE Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture*, 1149.7-2009 |
| USB xHCI Specification | 1.1 | Dec. 20, 2013 | *eXtensible Host Controller Interface for Universal Serial Bus (xHCI) Specification*, Revision 1.1 |

## 38.1 Signal Descriptions

The signal descriptions are shown in Table 38-2, "Signal Names and Descriptions." For additional signal information, including whether a signal is assigned to shared-function pin or can be used as a Customer GPIO signals see Chapter 39, "Signal Pin Names and Signal MUXing." The Direction/Type column of Table 38-2, "Signal Names and Descriptions" is interpreted as follows:

- I = Input signal. The SoC has receiver circuit.
- O = Output signal. The SoC has driver circuit.
- O-OD = Open-Drain Output. SoC output circuit uses an open-drain driver.

**Table 38-2. Signal Names and Descriptions**

| Signal Names | Direction | Shared | Description |
|---|---|---|---|
| CTBTRIGINOUT | I,O | Yes | **CTB Trigger:** Signals for triggering an external logic analyzer based on internal SoC events. |
| CTBTRIGOUT | O | Yes | |
| CX_PRDY_N | O-OD | Yes | **Debug Port Signals:** For connection to a MIPI-60 header on the board. |
| CX_PREQ_N | I | Yes | |
| DFX_PORT[15:0] | O | Yes | |
| DFX_PORT_CLK[1:0] | O | Yes | |
| **Test Access Port 0 - for Debug Port MIPI-60 Header on Board** | | | |
| TCK | I | No | **Test Clock:** For debug-capable platform board, an external 51-Ω +/- 5% resistor tied to VSS is required. Otherwise, no external resistor is required. |
| TDI | I | No | **Test Data Input:** For debug-capable platform board, an external 51-Ω +/- 5% resistor tied to V1P05 is required. Otherwise, no external resistor is required. |
| TDO | I,O-OD | No | **Test Data Output:** For debug-capable platform board, an external 51-Ω +/- 5% resistor tied to V1P05 is required. Otherwise, no external resistor is required. |
| TMS | I | No | **Test Mode Select:** For debug-capable platform board, an external 51-Ω +/- 5% resistor tied to V1P05 is required. Otherwise, no external resistor is required. |
| TRST_N | I | No | **Test Reset:** For debug-capable platform board, an external resistor within the range of 500-Ω to 1kΩ tied to VSS is required. Otherwise, no external resistor is required. |

## 38.2 Feature List

- Intel® Direct Connect Interface (Intel® DCI)
  - Debug Capability (DbC)
  - Intel® DCI-OOB (Out of Band)
- MIPI Parallel Trace Interface (PTI) capability
- IEEE Test Access Port (TAP)
- Intel® Trace Hub (Intel® TH)

## 38.3 Signals after Reset

- Following Power-up Reset, Warm Reset, and Cold Reset, all of the signals listed in Table 38-2, "Signal Names and Descriptions" are available at the SoC pins except for:
  - CTBTRIGINOUT
  - CTBTRIGOUT

To make these signals available, software must reconfigure these pins. The pin configuration mechanism is explained in Chapter 26, "Customer General-Purpose I/O (GPIO)."

## 38.4 Intel® Direct Connect Interface (Intel® DCI)

Intel® DCI provides a USB 3.0 path to debug features that can replace the traditional Test Access Port (TAP) pins and GPIO. Intel® DCI is being developed as an industry standard through the Mobile Industry Processor Interface Alliance (MIPI* Alliance) standards group. USB-hosted Intel® DCI provides TAP Access, Trace Extraction and Trace Capture.

Depending on the product SKU, lanes 19:16 are set to be either SATA or USB 3.0. For lane configurations, see Chapter 10, "Flexible I/O Adapter (FIA) Overview" For Intel® DCI to work correctly, the Intel® ME programs the Intel® DCI registers to indicate which lanes are SATA and which are USB 3.0 for the particular SoC product SKU.

Debug Capability (DbC) and Intel® DCI-OOB (Out of Band) are part of Intel® DCI. Any USB 3.0 lane (port) can be used for DbC or Intel® DCI-OOB. Two lanes are the maximum that can be used, only if they are in different modes (one lane for DbC and one lane for Intel® DCI-OOB) and used at different times. Intel® DCI does not support dual-mode (simultaneous) operation.

Whichever USB 3.0 port receives an external DbC request switches into the DbC mode and that is the only USB 3.0 port allowed to be in DbC mode until the port goes out of DbC mode.

### 38.4.1 Debug Capability (DbC)

The USB Debug Capability is a hosting transport that a debugger can use to control a platform. It is provided by the SoC USB eXtensible Host Controller (xHC). DbC is an optional USB functionality that enables low-level system debug over USB. The xHC Interface (xHCI) debugging capability provides a means of connecting two systems where one system is a Debug Host and the other a Debug Target (System Under Test).

The Host Memory Space range from 0xFED6_0000 to 0xFED6_0FFF is reserved for traffic targeting the SoC xHCI DbC subsystem. The Operating System must be informed that this address range is reserved. No other Memory-Mapped I/O (MMIO) in Host Memory Space must overlap with this address range. DbC is never a memory target. It always reads data from elsewhere and transfers in/out through the SoC USB 3.0 port.

See the section on DbC in the *eXtensible Host Controller Interface for Universal Serial Bus (xHCI), Revision 1.1*.

### 38.4.2 Intel® DCI-OOB (Out of Band)

Intel® DCI-OOB (Out of Band) is a hosting transport that a debugger can use to control a platform. Intel® DCI-OOB is supported by the SoC but requires a special USB 3.0 to Intel® DCI-OOB adapter. Intel® DCI-OOB gives the SoC customer the ability to do the same debug capabilities as DbC, but without the restriction of running it while the operating system (OS) is running on the target. It can run at any time, even during reset.

Intel® DCI-OOB has lower bandwidth when compared to DbC and only supports two of three main operations. Intel® DCI-OOB supports TAP control and Trace Streaming from Intel® Trace Hub, but it does not support memory extraction from main memory. Only DbC supports memory extraction from main memory.

## 38.5 Intel® Trace Hub

While Intel® Direct Connect Interface (Intel® DCI) is for closed chassis debug, the Intel® Trace Hub (Intel® TH) is an integrated PCI device for system tracing debug. It is discovered as Bus 0, Device 31 (decimal), Function 7.

The Intel® TH is an integrated trace analyzer and a set of silicon features allowing the user to perform system-level debug. It provides hardware-software co-debug. Intel is enabling third-party debug-software vendors to use the integrated Intel® TH in their products to unify software and hardware visibility and internal tracing capabilities.

An Intel® TH viewer is bundled with the Intel® System Studio product to allow the user one common viewer for hardware and software co-development.

The Intel® TH allows the user to trigger sources and capture the events into system memory (DRAM). The user can retrieve the events stored in memory for viewing with the software compiler debug tool (ISS) or retrieve the events from system memory through a third-party vendor JTAG tool using industry-standard MIPI (STPv2) interface and the Parallel Trace Interface (PTI).

## 38.6 On-Board Debug Port Connector

The SoC provides debug-related signals that can be accessed by a 60-pin Mobile Industry Processor Interface (MIPI) connector. MIPI calls this the Parallel Trace Interface (PTI).

§ §

# Volume 3: Electrical, Mechanical, and Thermal



February 2018
Order Number: 337018-002

# 39 Signal Pin Names and Signal MUXing

***Warning:*** The information in this chapter is preliminary and is subject to change.

The SoC pin numbers, signal-pin names, and associated signals are listed in Table 39-1, "Signal Pin Name Directory - Sorted by Name" in alphabetical order by signal-pin name. Some pins shown in the table have multiple uses and are shared by multiple signals.

The voltage supply pins and voltage sense pins are in a separate list, Table 39-4, "Voltage Pins Sorted by Platform VR and SoC Voltage Group."

Table 39-6, "Directory of Signal and Voltage Pins - Sorted by Pin Number" lists all of the SoC pins sorted by pin number.

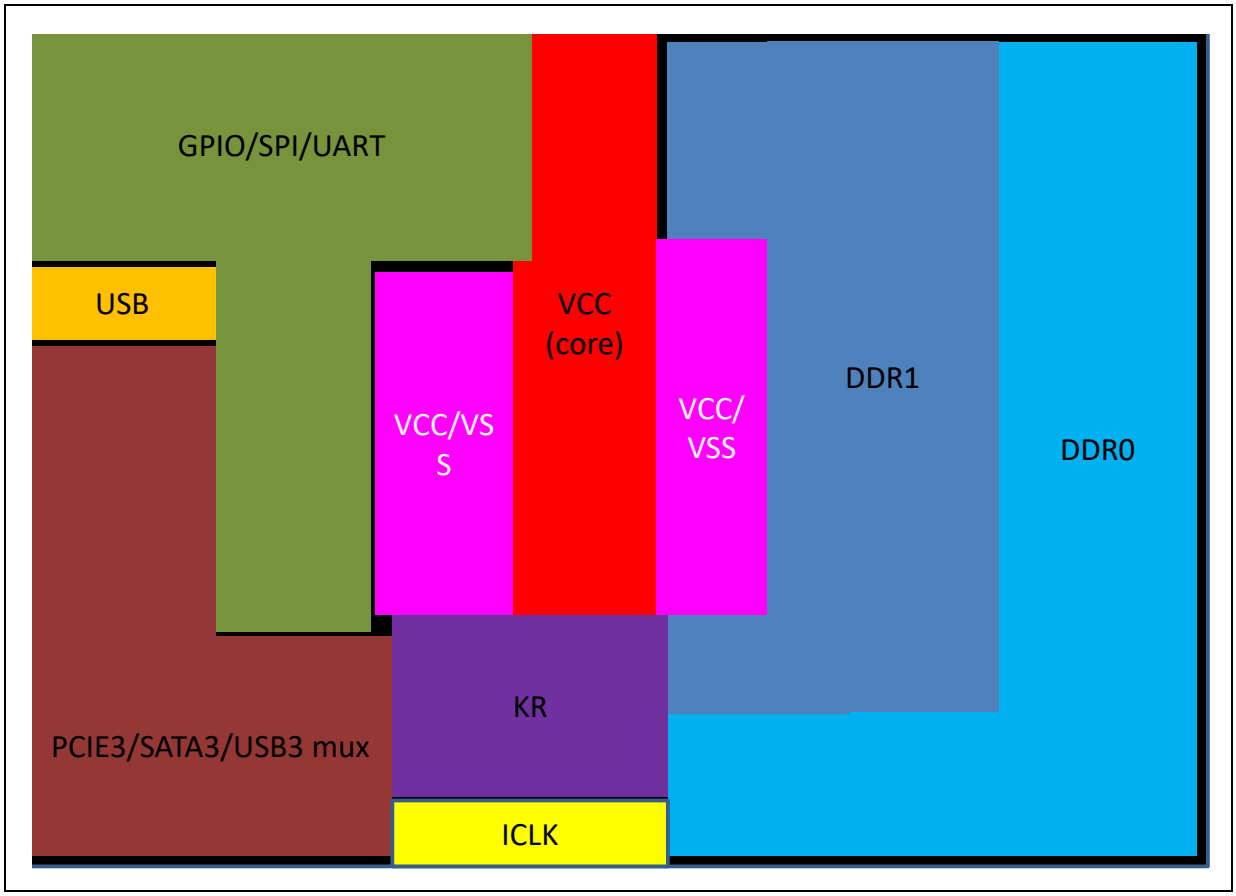
The Transmit and Receive differential interface signals for the lanes of the integrated PCI Express* Root Ports, SATA controller ports, and USB 3.0 ports share up to 20 High-Speed I/O (HSIO) lanes of the SoC. These are shown in Table 39-3, "Signals that Share the Twenty HSIO Lanes."

See also Chapter 10, "Flexible I/O Adapter (FIA) Overview."

*Note:* All signal descriptions are in the functional sections of this document. Readers of the electronic version of this document can navigate to the signal descriptions by clicking on the blue-lettered "D" in the Description column of the following tables:

- Table 39-1, "Signal Pin Name Directory - Sorted by Name"
- Table 39-3, "Signals that Share the Twenty HSIO Lanes."

**Figure 39-1. Ball Quadrant Map**



## 39.1 Directory of Signal Names and Pin Names

The table headers for Table 39-1, "Signal Pin Name Directory - Sorted by Name":

- Pin - SoC Pin/Ball matrix number
- Pin Name - Signal Name assigned to the SoC Pin/Ball
- Buffer Type - The type of driver/receiver circuitry used by the signals associated with the SoC Pin/Ball.
- V Group - SoC Voltage Group that supplies power to the driver/receiver circuitry.
- Pin States - Shows the possible states of what is connected to the pin:
  - Not Shared: The signal is not shared. None of the pin states listed below apply to the pin.
  - Strap Sampling: Shown if the pin serves as a Hard Strap during the SoC Hard-Strap pin sampling period. The sampling period is described in Section 4.1, "SoC Hard (Pin) Straps."
  - GPIO Mode: Also known as Func 0, is shown if the pin is capable of being programmed to be a Customer GPIO.
  - Default: The default internal signal connection to the pin after reset. In some cases, the pin's default setting is the Customer GPIO for the pin. Also, in some cases, The LAN Controller firmware can alter a signal-pin's default setting just before BIOS starts.
  - Func x: For pins that are shared by multiple signals, the Function number (Func 1, Func 2,..., Func 7) is programmable by BIOS. There can be up to seven different functional signals per shared pin, but in most cases, less than seven are used. The default pin-MUX setting is the pin's signal after a reset and is not necessarily Func 1. In some cases the default is a reset-configured Customer GPIO for the pin.
- Signal Name - The name of the signal internally connected to the pin for the given Pin State.
- Dir - The direction of the signal and whether the signal is an open-drain, tri-state, differential, or analog circuit:
  - I: SoC Input
  - O: SoC Output
  - O-OD: SoC Open-Drain Output
  - O-Tri: SoC Tri-State Output (logic 0, logic 1, and Hi-Z states)
  - Diff: Differential circuit
  - Analog: Not a digital-logic signal. Typically associated with pins for external compensation circuits or pins connected to internal thermal-sensing diodes.
  - SW: The PU/PD is programmed by BIOS. Typically, this is meant for Customer GPIOs.

- Internal PU/PD - Shown where the SoC has an internal Pull-Up (PU) resistor, Pull-Down (PD) resistor, or both, connected to the pin.
  - The Internal PU/PD shown for the default setting (As BIOS Starts state) remains intact even after BIOS alters the Func 1, Func 2, etc. setting.
  - The internal termination can be altered by BIOS which can program a new PU and/or PD for the pin buffer circuitry or remove all internal resistor connections to the pin.
  - For pins designated as Hard Straps, the SoC applies a special PU/PD arrangement that is active only during the strap-sensing period. After this sampling period, the default Internal PU/PD is applied to the pin.
- Signal Descriptions - Readers of the electronic version of this document can navigate to the chapter containing the signal description by clicking on the blue-colored "D" in the Description column of Table 39-1, "Signal Pin Name Directory - Sorted by Name."

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 1 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BV60 | ADR_TRIGGER | CMOS-HV | V3P30 | Default | GPIO[81] | O | - | D |
| | | | | Func 0 GPIO | GPIO[81] | SW | SW | D |
| | | | | Func 1 | ADR_TRIGGER | I,O-OD | - | D |
| AU61 | BVCCRTC_EXTPAD | RTC | VRTC | not shared | BVCCRTC_EXTPAD | I,O | - | D |
| A41 | CLK_OUT_DN[0] | ISCLK | V1P05 | not shared | CLK_OUT_DN[0] | O, Diff | - | D |
| A43 | CLK_OUT_DN[1] | ISCLK | V1P05 | not shared | CLK_OUT_DN[1] | O, Diff | - | D |
| A45 | CLK_OUT_DN[2] | ISCLK | V1P05 | not shared | CLK_OUT_DN[2] | O, Diff | - | D |
| A48 | CLK_OUT_DN[3] | ISCLK | V1P05 | not shared | CLK_OUT_DN[3] | O, Diff | - | D |
| A50 | CLK_OUT_DN[4] | ISCLK | V1P05 | not shared | CLK_OUT_DN[4] | O, Diff | - | D |
| C41 | CLK_OUT_DP[0] | ISCLK | V1P05 | not shared | CLK_OUT_DP[0] | O, Diff | - | D |
| C43 | CLK_OUT_DP[1] | ISCLK | V1P05 | not shared | CLK_OUT_DP[1] | O, Diff | - | D |
| C45 | CLK_OUT_DP[2] | ISCLK | V1P05 | not shared | CLK_OUT_DP[2] | O, Diff | - | D |
| C48 | CLK_OUT_DP[3] | ISCLK | V1P05 | not shared | CLK_OUT_DP[3] | O, Diff | - | D |
| C50 | CLK_OUT_DP[4] | ISCLK | V1P05 | not shared | CLK_OUT_DP[4] | O, Diff | - | D |
| C39 | CLK_X1_PAD | KR | V1P05 | not shared | CLK_X1_PAD | I | - | D |
| A39 | CLK_X2_PAD | KR | V1P05 | not shared | CLK_X2_PAD | O | - | D |
| AG56 | COREPWROK | RTC | VRTC | not shared | COREPWROK | I | - | D |
| BE44 | CTBTRIGINOUT | CMOS-LV | V1P05 | Default | Reserved | I | 20K PU | |
| | | | | Func 0 GPIO | GPIO[148] | SW | SW | D |
| | | | | Func 1 | Reserved | I | 20K PU | |
| | | | | Func 2 | CTBTRIGINOUT | I,O | 20K PU | D |
| | | | | Func 3 | CTBTRIGINOUT | I,O | 20K PU | |
| CA39 | CTBTRIGOUT | CMOS-LV | V1P05 | Default | Reserved | I | - | |
| | | | | Func 0 GPIO | GPIO[149] | SW | SW | D |
| | | | | Func 1 | Reserved | I | - | |
| | | | | Func 2 | CTBTRIGOUT | O | - | D |
| | | | | Func 3 | Reserved | O | - | |
| BY38 | CX_PRDY_N | CMOS-LV | V1P05 | Default | CX_PRDY_N | O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[146] | SW | SW | D |
| | | | | Func 1 | CX_PRDY_N | O-OD | 20K PU | D |
| BH42 | CX_PREQ_N | CMOS-LV | V1P05 | Default | CX_PREQ_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[147] | SW | SW | D |
| | | | | Func 1 | CX_PREQ_N | I | 20K PU | D |
| AV3 | DDR0_ACT_N_MA[15] | DDR | VDDQ | not shared | DDR0_ACT_N_MA[15] | O | - | D |
| AY3 | DDR0_ALERT_N_PAR_ERR_N | DDR | VDDQ | not shared | DDR0_ALERT_N_PAR_ERR_N | I | None | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 2 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BV8 | DDR0_BA[0] | DDR | VDDQ | not shared | DDR0_BA[0] | O | - | D |
| BT5 | DDR0_BA[1] | DDR | VDDQ | not shared | DDR0_BA[1] | O | - | D |
| AW4 | DDR0_BG0_BA[2] | DDR | VDDQ | not shared | DDR0_BG0_BA[2] | O | - | D |
| AW2 | DDR0_BG1_MA[14] | DDR | VDDQ | not shared | DDR0_BG1_MA[14] | O | - | D |
| BV14 | DDR0_C0 | DDR | VDDQ | not shared | DDR0_C0 | O | - | D |
| AT4 | DDR0_CKE[0] | DDR | VDDQ | not shared | DDR0_CKE[0] | O | - | D |
| AP4 | DDR0_CKE[1] | DDR | VDDQ | not shared | DDR0_CKE[1] | O | - | D |
| AT2 | DDR0_CKE[2] | DDR | VDDQ | not shared | DDR0_CKE[2] | O | - | D |
| AR3 | DDR0_CKE[3] | DDR | VDDQ | not shared | DDR0_CKE[3] | O | - | D |
| BM4 | DDR0_CLK_DN[0] | DDR | VDDQ | not shared | DDR0_CLK_DN[0] | O, Diff | - | D |
| BK4 | DDR0_CLK_DN[1] | DDR | VDDQ | not shared | DDR0_CLK_DN[1] | O, Diff | - | D |
| BL3 | DDR0_CLK_DN[2] | DDR | VDDQ | not shared | DDR0_CLK_DN[2] | O, Diff | - | D |
| BH4 | DDR0_CLK_DN[3] | DDR | VDDQ | not shared | DDR0_CLK_DN[3] | O, Diff | - | D |
| BM2 | DDR0_CLK_DP[0] | DDR | VDDQ | not shared | DDR0_CLK_DP[0] | O, Diff | - | D |
| BK2 | DDR0_CLK_DP[1] | DDR | VDDQ | not shared | DDR0_CLK_DP[1] | O, Diff | - | D |
| BL1 | DDR0_CLK_DP[2] | DDR | VDDQ | not shared | DDR0_CLK_DP[2] | O, Diff | - | D |
| BH2 | DDR0_CLK_DP[3] | DDR | VDDQ | not shared | DDR0_CLK_DP[3] | O, Diff | - | D |
| BW5 | DDR0_CS_N[0] | DDR | VDDQ | not shared | DDR0_CS_N[0] | O | - | D |
| BV12 | DDR0_CS_N[1] | DDR | VDDQ | not shared | DDR0_CS_N[1] | O | - | D |
| BV10 | DDR0_CS_N[2] | DDR | VDDQ | not shared | DDR0_CS_N[2] | O | - | D |
| BW15 | DDR0_CS_N[3] | DDR | VDDQ | not shared | DDR0_CS_N[3] | O | - | D |
| B12 | DDR0_DQ[0] | DDR | VDDQ | not shared | DDR0_DQ[0] | I,O | - | D |
| D12 | DDR0_DQ[1] | DDR | VDDQ | not shared | DDR0_DQ[1] | I,O | - | D |
| C9 | DDR0_DQ[2] | DDR | VDDQ | not shared | DDR0_DQ[2] | I,O | - | D |
| E6 | DDR0_DQ[3] | DDR | VDDQ | not shared | DDR0_DQ[3] | I,O | - | D |
| A13 | DDR0_DQ[4] | DDR | VDDQ | not shared | DDR0_DQ[4] | I,O | - | D |
| C13 | DDR0_DQ[5] | DDR | VDDQ | not shared | DDR0_DQ[5] | I,O | - | D |
| D8 | DDR0_DQ[6] | DDR | VDDQ | not shared | DDR0_DQ[6] | I,O | - | D |
| C6 | DDR0_DQ[7] | DDR | VDDQ | not shared | DDR0_DQ[7] | I,O | - | D |
| M2 | DDR0_DQ[8] | DDR | VDDQ | not shared | DDR0_DQ[8] | I,O | - | D |
| M4 | DDR0_DQ[9] | DDR | VDDQ | not shared | DDR0_DQ[9] | I,O | - | D |
| T2 | DDR0_DQ[10] | DDR | VDDQ | not shared | DDR0_DQ[10] | I,O | - | D |
| T4 | DDR0_DQ[11] | DDR | VDDQ | not shared | DDR0_DQ[11] | I,O | - | D |
| L1 | DDR0_DQ[12] | DDR | VDDQ | not shared | DDR0_DQ[12] | I,O | - | D |
| L3 | DDR0_DQ[13] | DDR | VDDQ | not shared | DDR0_DQ[13] | I,O | - | D |
| R1 | DDR0_DQ[14] | DDR | VDDQ | not shared | DDR0_DQ[14] | I,O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 3 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| R3 | DDR0_DQ[15] | DDR | VDDQ | not shared | DDR0_DQ[15] | I,O | - | D |
| Y1 | DDR0_DQ[16] | DDR | VDDQ | not shared | DDR0_DQ[16] | I,O | - | D |
| Y3 | DDR0_DQ[17] | DDR | VDDQ | not shared | DDR0_DQ[17] | I,O | - | D |
| AD1 | DDR0_DQ[18] | DDR | VDDQ | not shared | DDR0_DQ[18] | I,O | - | D |
| AD3 | DDR0_DQ[19] | DDR | VDDQ | not shared | DDR0_DQ[19] | I,O | - | D |
| W2 | DDR0_DQ[20] | DDR | VDDQ | not shared | DDR0_DQ[20] | I,O | - | D |
| W4 | DDR0_DQ[21] | DDR | VDDQ | not shared | DDR0_DQ[21] | I,O | - | D |
| AC2 | DDR0_DQ[22] | DDR | VDDQ | not shared | DDR0_DQ[22] | I,O | - | D |
| AC4 | DDR0_DQ[23] | DDR | VDDQ | not shared | DDR0_DQ[23] | I,O | - | D |
| AC11 | DDR0_DQ[24] | DDR | VDDQ | not shared | DDR0_DQ[24] | I,O | - | D |
| AC7 | DDR0_DQ[25] | DDR | VDDQ | not shared | DDR0_DQ[25] | I,O | - | D |
| AJ7 | DDR0_DQ[26] | DDR | VDDQ | not shared | DDR0_DQ[26] | I,O | - | D |
| AJ11 | DDR0_DQ[27] | DDR | VDDQ | not shared | DDR0_DQ[27] | I,O | - | D |
| AA9 | DDR0_DQ[28] | DDR | VDDQ | not shared | DDR0_DQ[28] | I,O | - | D |
| AD9 | DDR0_DQ[29] | DDR | VDDQ | not shared | DDR0_DQ[29] | I,O | - | D |
| AH9 | DDR0_DQ[30] | DDR | VDDQ | not shared | DDR0_DQ[30] | I,O | - | D |
| AG11 | DDR0_DQ[31] | DDR | VDDQ | not shared | DDR0_DQ[31] | I,O | - | D |
| BY19 | DDR0_DQ[32] | DDR | VDDQ | not shared | DDR0_DQ[32] | I,O | - | D |
| BV19 | DDR0_DQ[33] | DDR | VDDQ | not shared | DDR0_DQ[33] | I,O | - | D |
| BY23 | DDR0_DQ[34] | DDR | VDDQ | not shared | DDR0_DQ[34] | I,O | - | D |
| BV23 | DDR0_DQ[35] | DDR | VDDQ | not shared | DDR0_DQ[35] | I,O | - | D |
| CA18 | DDR0_DQ[36] | DDR | VDDQ | not shared | DDR0_DQ[36] | I,O | - | D |
| BW18 | DDR0_DQ[37] | DDR | VDDQ | not shared | DDR0_DQ[37] | I,O | - | D |
| CA22 | DDR0_DQ[38] | DDR | VDDQ | not shared | DDR0_DQ[38] | I,O | - | D |
| BW22 | DDR0_DQ[39] | DDR | VDDQ | not shared | DDR0_DQ[39] | I,O | - | D |
| BG22 | DDR0_DQ[40] | DDR | VDDQ | not shared | DDR0_DQ[40] | I,O | - | D |
| BE22 | DDR0_DQ[41] | DDR | VDDQ | not shared | DDR0_DQ[41] | I,O | - | D |
| BG29 | DDR0_DQ[42] | DDR | VDDQ | not shared | DDR0_DQ[42] | I,O | - | D |
| BE29 | DDR0_DQ[43] | DDR | VDDQ | not shared | DDR0_DQ[43] | I,O | - | D |
| BH20 | DDR0_DQ[44] | DDR | VDDQ | not shared | DDR0_DQ[44] | I,O | - | D |
| BF20 | DDR0_DQ[45] | DDR | VDDQ | not shared | DDR0_DQ[45] | I,O | - | D |
| BH27 | DDR0_DQ[46] | DDR | VDDQ | not shared | DDR0_DQ[46] | I,O | - | D |
| BF27 | DDR0_DQ[47] | DDR | VDDQ | not shared | DDR0_DQ[47] | I,O | - | D |
| CA26 | DDR0_DQ[48] | DDR | VDDQ | not shared | DDR0_DQ[48] | I,O | - | D |
| BW26 | DDR0_DQ[49] | DDR | VDDQ | not shared | DDR0_DQ[49] | I,O | - | D |
| CA30 | DDR0_DQ[50] | DDR | VDDQ | not shared | DDR0_DQ[50] | I,O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 4 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BW30 | DDR0_DQ[51] | DDR | VDDQ | not shared | DDR0_DQ[51] | I,O | - | D |
| BY25 | DDR0_DQ[52] | DDR | VDDQ | not shared | DDR0_DQ[52] | I,O | - | D |
| BV25 | DDR0_DQ[53] | DDR | VDDQ | not shared | DDR0_DQ[53] | I,O | - | D |
| BY29 | DDR0_DQ[54] | DDR | VDDQ | not shared | DDR0_DQ[54] | I,O | - | D |
| BV29 | DDR0_DQ[55] | DDR | VDDQ | not shared | DDR0_DQ[55] | I,O | - | D |
| BN22 | DDR0_DQ[56] | DDR | VDDQ | not shared | DDR0_DQ[56] | I,O | - | D |
| BL22 | DDR0_DQ[57] | DDR | VDDQ | not shared | DDR0_DQ[57] | I,O | - | D |
| BN29 | DDR0_DQ[58] | DDR | VDDQ | not shared | DDR0_DQ[58] | I,O | - | D |
| BL29 | DDR0_DQ[59] | DDR | VDDQ | not shared | DDR0_DQ[59] | I,O | - | D |
| BR20 | DDR0_DQ[60] | DDR | VDDQ | not shared | DDR0_DQ[60] | I,O | - | D |
| BM20 | DDR0_DQ[61] | DDR | VDDQ | not shared | DDR0_DQ[61] | I,O | - | D |
| BR27 | DDR0_DQ[62] | DDR | VDDQ | not shared | DDR0_DQ[62] | I,O | - | D |
| BM27 | DDR0_DQ[63] | DDR | VDDQ | not shared | DDR0_DQ[63] | I,O | - | D |
| B10 | DDR0_DQS_DN[0] | DDR | VDDQ | not shared | DDR0_DQS_DN[0] | I,O, Diff | - | D |
| P2 | DDR0_DQS_DN[1] | DDR | VDDQ | not shared | DDR0_DQS_DN[1] | I,O, Diff | - | D |
| AB1 | DDR0_DQS_DN[2] | DDR | VDDQ | not shared | DDR0_DQS_DN[2] | I,O, Diff | - | D |
| AE7 | DDR0_DQS_DN[3] | DDR | VDDQ | not shared | DDR0_DQS_DN[3] | I,O, Diff | - | D |
| BY21 | DDR0_DQS_DN[4] | DDR | VDDQ | not shared | DDR0_DQS_DN[4] | I,O, Diff | - | D |
| BG26 | DDR0_DQS_DN[5] | DDR | VDDQ | not shared | DDR0_DQS_DN[5] | I,O, Diff | - | D |
| CA28 | DDR0_DQS_DN[6] | DDR | VDDQ | not shared | DDR0_DQS_DN[6] | I,O, Diff | - | D |
| BN26 | DDR0_DQS_DN[7] | DDR | VDDQ | not shared | DDR0_DQS_DN[7] | I,O, Diff | - | D |
| AK2 | DDR0_DQS_DN[8] | DDR | VDDQ | not shared | DDR0_DQS_DN[8] | I,O, Diff | - | D |
| C11 | DDR0_DQS_DN[9] | DDR | VDDQ | not shared | DDR0_DQS_DN[9] | I,O, Diff | - | D |
| N3 | DDR0_DQS_DN[10] | DDR | VDDQ | not shared | DDR0_DQS_DN[10] | I,O, Diff | - | D |
| AA4 | DDR0_DQS_DN[11] | DDR | VDDQ | not shared | DDR0_DQS_DN[11] | I,O, Diff | - | D |
| AE11 | DDR0_DQS_DN[12] | DDR | VDDQ | not shared | DDR0_DQS_DN[12] | I,O, Diff | - | D |
| BW20 | DDR0_DQS_DN[13] | DDR | VDDQ | not shared | DDR0_DQS_DN[13] | I,O, Diff | - | D |
| BF24 | DDR0_DQS_DN[14] | DDR | VDDQ | not shared | DDR0_DQS_DN[14] | I,O, Diff | - | D |
| BV27 | DDR0_DQS_DN[15] | DDR | VDDQ | not shared | DDR0_DQS_DN[15] | I,O, Diff | - | D |
| BM24 | DDR0_DQS_DN[16] | DDR | VDDQ | not shared | DDR0_DQS_DN[16] | I,O, Diff | - | D |
| AJ3 | DDR0_DQS_DN[17] | DDR | VDDQ | not shared | DDR0_DQS_DN[17] | I,O, Diff | - | D |
| D10 | DDR0_DQS_DP[0] | DDR | VDDQ | not shared | DDR0_DQS_DP[0] | I,O, Diff | - | D |
| P4 | DDR0_DQS_DP[1] | DDR | VDDQ | not shared | DDR0_DQS_DP[1] | I,O, Diff | - | D |
| AB3 | DDR0_DQS_DP[2] | DDR | VDDQ | not shared | DDR0_DQS_DP[2] | I,O, Diff | - | D |
| AG7 | DDR0_DQS_DP[3] | DDR | VDDQ | not shared | DDR0_DQS_DP[3] | I,O, Diff | - | D |
| BV21 | DDR0_DQS_DP[4] | DDR | VDDQ | not shared | DDR0_DQS_DP[4] | I,O, Diff | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 5 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BE26 | DDR0_DQS_DP[5] | DDR | VDDQ | not shared | DDR0_DQS_DP[5] | I,O, Diff | - | D |
| BW28 | DDR0_DQS_DP[6] | DDR | VDDQ | not shared | DDR0_DQS_DP[6] | I,O, Diff | - | D |
| BL26 | DDR0_DQS_DP[7] | DDR | VDDQ | not shared | DDR0_DQS_DP[7] | I,O, Diff | - | D |
| AK4 | DDR0_DQS_DP[8] | DDR | VDDQ | not shared | DDR0_DQS_DP[8] | I,O, Diff | - | D |
| A11 | DDR0_DQS_DP[9] | DDR | VDDQ | not shared | DDR0_DQS_DP[9] | I,O, Diff | - | D |
| N1 | DDR0_DQS_DP[10] | DDR | VDDQ | not shared | DDR0_DQS_DP[10] | I,O, Diff | - | D |
| AA2 | DDR0_DQS_DP[11] | DDR | VDDQ | not shared | DDR0_DQS_DP[11] | I,O, Diff | - | D |
| AF9 | DDR0_DQS_DP[12] | DDR | VDDQ | not shared | DDR0_DQS_DP[12] | I,O, Diff | - | D |
| CA20 | DDR0_DQS_DP[13] | DDR | VDDQ | not shared | DDR0_DQS_DP[13] | I,O, Diff | - | D |
| BH24 | DDR0_DQS_DP[14] | DDR | VDDQ | not shared | DDR0_DQS_DP[14] | I,O, Diff | - | D |
| BY27 | DDR0_DQS_DP[15] | DDR | VDDQ | not shared | DDR0_DQS_DP[15] | I,O, Diff | - | D |
| BR24 | DDR0_DQS_DP[16] | DDR | VDDQ | not shared | DDR0_DQS_DP[16] | I,O, Diff | - | D |
| AJ1 | DDR0_DQS_DP[17] | DDR | VDDQ | not shared | DDR0_DQS_DP[17] | I,O, Diff | - | D |
| AH2 | DDR0_ECC[0] | DDR | VDDQ | not shared | DDR0_ECC[0] | I,O | - | D |
| AH4 | DDR0_ECC[1] | DDR | VDDQ | not shared | DDR0_ECC[1] | I,O | - | D |
| AM2 | DDR0_ECC[2] | DDR | VDDQ | not shared | DDR0_ECC[2] | I,O | - | D |
| AM4 | DDR0_ECC[3] | DDR | VDDQ | not shared | DDR0_ECC[3] | I,O | - | D |
| AF1 | DDR0_ECC[4] | DDR | VDDQ | not shared | DDR0_ECC[4] | I,O | - | D |
| AF3 | DDR0_ECC[5] | DDR | VDDQ | not shared | DDR0_ECC[5] | I,O | - | D |
| AL1 | DDR0_ECC[6] | DDR | VDDQ | not shared | DDR0_ECC[6] | I,O | - | D |
| AL3 | DDR0_ECC[7] | DDR | VDDQ | not shared | DDR0_ECC[7] | I,O | - | D |
| BP4 | DDR0_MA[0] | DDR | VDDQ | not shared | DDR0_MA[0] | O | - | D |
| BF3 | DDR0_MA[1] | DDR | VDDQ | not shared | DDR0_MA[1] | O | - | D |
| BF1 | DDR0_MA[2] | DDR | VDDQ | not shared | DDR0_MA[2] | O | - | D |
| BE2 | DDR0_MA[3] | DDR | VDDQ | not shared | DDR0_MA[3] | O | - | D |
| BE4 | DDR0_MA[4] | DDR | VDDQ | not shared | DDR0_MA[4] | O | - | D |
| BD3 | DDR0_MA[5] | DDR | VDDQ | not shared | DDR0_MA[5] | O | - | D |
| BC2 | DDR0_MA[6] | DDR | VDDQ | not shared | DDR0_MA[6] | O | - | D |
| BB1 | DDR0_MA[7] | DDR | VDDQ | not shared | DDR0_MA[7] | O | - | D |
| BC4 | DDR0_MA[8] | DDR | VDDQ | not shared | DDR0_MA[8] | O | - | D |
| BA2 | DDR0_MA[9] | DDR | VDDQ | not shared | DDR0_MA[9] | O | - | D |
| BU6 | DDR0_MA[10] | DDR | VDDQ | not shared | DDR0_MA[10] | O | - | D |
| BB3 | DDR0_MA[11] | DDR | VDDQ | not shared | DDR0_MA[11] | O | - | D |
| BA4 | DDR0_MA[12] | DDR | VDDQ | not shared | DDR0_MA[12] | O | - | D |
| BW11 | DDR0_MA[13] | DDR | VDDQ | not shared | DDR0_MA[13] | O | - | D |
| BW9 | DDR0_MA14_WE_N | DDR | VDDQ | not shared | DDR0_MA14_WE_N | O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 6 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| CA11 | DDR0_MA15_CAS_N | DDR | VDDQ | not shared | DDR0_MA15_CAS_N | O | - | D |
| BW6 | DDR0_MA16_RAS_N | DDR | VDDQ | not shared | DDR0_MA16_RAS_N | O | - | D |
| CA13 | DDR0_MA17 | DDR | VDDQ | not shared | DDR0_MA17 | O | - | D |
| BY10 | DDR0_ODT[0] | DDR | VDDQ | not shared | DDR0_ODT[0] | O | - | D |
| BY14 | DDR0_ODT[1] | DDR | VDDQ | not shared | DDR0_ODT[1] | O | - | D |
| BW13 | DDR0_ODT[2] | DDR | VDDQ | not shared | DDR0_ODT[2] | O | - | D |
| CA15 | DDR0_ODT[3] | DDR | VDDQ | not shared | DDR0_ODT[3] | O | - | D |
| BN3 | DDR0_PAR | DDR | VDDQ | not shared | DDR0_PAR | O | - | D |
| AD24 | DDR01_COMP | DDR | VDDQ | not shared | DDR01_COMP | O | - | D |
| AP2 | DDR01_DRAMRST_N | DDR | VDDQ | not shared | DDR01_DRAMRST_N | O | - | D |
| H9 | DDR1_ACT_N_MA[15] | DDR | VDDQ | not shared | DDR1_ACT_N_MA[15] | O | - | D |
| M15 | DDR1_ALERT_N_PAR_ERR_N | DDR | VDDQ | not shared | DDR1_ALERT_N_PAR_ERR_N | I | None | D |
| AD20 | DDR1_BA[0] | DDR | VDDQ | not shared | DDR1_BA[0] | O | - | D |
| AA20 | DDR1_BA[1] | DDR | VDDQ | not shared | DDR1_BA[1] | O | - | D |
| G7 | DDR1_BG0_BA[2] | DDR | VDDQ | not shared | DDR1_BG0_BA[2] | O | - | D |
| J5 | DDR1_BG1_MA[14] | DDR | VDDQ | not shared | DDR1_BG1_MA[14] | O | - | D |
| AJ15 | DDR1_C0 | DDR | VDDQ | not shared | DDR1_C0 | O | - | D |
| K13 | DDR1_CKE[0] | DDR | VDDQ | not shared | DDR1_CKE[0] | O | - | D |
| J15 | DDR1_CKE[1] | DDR | VDDQ | not shared | DDR1_CKE[1] | O | - | D |
| H13 | DDR1_CKE[2] | DDR | VDDQ | not shared | DDR1_CKE[2] | O | - | D |
| G15 | DDR1_CKE[3] | DDR | VDDQ | not shared | DDR1_CKE[3] | O | - | D |
| AC15 | DDR1_CLK_DN[0] | DDR | VDDQ | not shared | DDR1_CLK_DN[0] | O, Diff | - | D |
| Y15 | DDR1_CLK_DN[1] | DDR | VDDQ | not shared | DDR1_CLK_DN[1] | O, Diff | - | D |
| V15 | DDR1_CLK_DN[2] | DDR | VDDQ | not shared | DDR1_CLK_DN[2] | O, Diff | - | D |
| U13 | DDR1_CLK_DN[3] | DDR | VDDQ | not shared | DDR1_CLK_DN[3] | O, Diff | - | D |
| AD17 | DDR1_CLK_DP[0] | DDR | VDDQ | not shared | DDR1_CLK_DP[0] | O, Diff | - | D |
| AA17 | DDR1_CLK_DP[1] | DDR | VDDQ | not shared | DDR1_CLK_DP[1] | O, Diff | - | D |
| W17 | DDR1_CLK_DP[2] | DDR | VDDQ | not shared | DDR1_CLK_DP[2] | O, Diff | - | D |
| T15 | DDR1_CLK_DP[3] | DDR | VDDQ | not shared | DDR1_CLK_DP[3] | O, Diff | - | D |
| AG22 | DDR1_CS_N[0] | DDR | VDDQ | not shared | DDR1_CS_N[0] | O | - | D |
| AV22 | DDR1_CS_N[1] | DDR | VDDQ | not shared | DDR1_CS_N[1] | O | - | D |
| AF17 | DDR1_CS_N[2] | DDR | VDDQ | not shared | DDR1_CS_N[2] | O | - | D |
| AH17 | DDR1_CS_N[3] | DDR | VDDQ | not shared | DDR1_CS_N[3] | O | - | D |
| A34 | DDR1_DQ[0] | DDR | VDDQ | not shared | DDR1_DQ[0] | I,O | - | D |
| C34 | DDR1_DQ[1] | DDR | VDDQ | not shared | DDR1_DQ[1] | I,O | - | D |
| A30 | DDR1_DQ[2] | DDR | VDDQ | not shared | DDR1_DQ[2] | I,O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 7 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| C30 | DDR1_DQ[3] | DDR | VDDQ | not shared | DDR1_DQ[3] | I,O | - | D |
| B35 | DDR1_DQ[4] | DDR | VDDQ | not shared | DDR1_DQ[4] | I,O | - | D |
| D35 | DDR1_DQ[5] | DDR | VDDQ | not shared | DDR1_DQ[5] | I,O | - | D |
| B31 | DDR1_DQ[6] | DDR | VDDQ | not shared | DDR1_DQ[6] | I,O | - | D |
| D31 | DDR1_DQ[7] | DDR | VDDQ | not shared | DDR1_DQ[7] | I,O | - | D |
| P26 | DDR1_DQ[8] | DDR | VDDQ | not shared | DDR1_DQ[8] | I,O | - | D |
| T26 | DDR1_DQ[9] | DDR | VDDQ | not shared | DDR1_DQ[9] | I,O | - | D |
| P18 | DDR1_DQ[10] | DDR | VDDQ | not shared | DDR1_DQ[10] | I,O | - | D |
| T18 | DDR1_DQ[11] | DDR | VDDQ | not shared | DDR1_DQ[11] | I,O | - | D |
| R27 | DDR1_DQ[12] | DDR | VDDQ | not shared | DDR1_DQ[12] | I,O | - | D |
| U27 | DDR1_DQ[13] | DDR | VDDQ | not shared | DDR1_DQ[13] | I,O | - | D |
| R20 | DDR1_DQ[14] | DDR | VDDQ | not shared | DDR1_DQ[14] | I,O | - | D |
| U20 | DDR1_DQ[15] | DDR | VDDQ | not shared | DDR1_DQ[15] | I,O | - | D |
| G26 | DDR1_DQ[16] | DDR | VDDQ | not shared | DDR1_DQ[16] | I,O | - | D |
| J26 | DDR1_DQ[17] | DDR | VDDQ | not shared | DDR1_DQ[17] | I,O | - | D |
| G18 | DDR1_DQ[18] | DDR | VDDQ | not shared | DDR1_DQ[18] | I,O | - | D |
| J18 | DDR1_DQ[19] | DDR | VDDQ | not shared | DDR1_DQ[19] | I,O | - | D |
| H27 | DDR1_DQ[20] | DDR | VDDQ | not shared | DDR1_DQ[20] | I,O | - | D |
| K27 | DDR1_DQ[21] | DDR | VDDQ | not shared | DDR1_DQ[21] | I,O | - | D |
| H20 | DDR1_DQ[22] | DDR | VDDQ | not shared | DDR1_DQ[22] | I,O | - | D |
| K20 | DDR1_DQ[23] | DDR | VDDQ | not shared | DDR1_DQ[23] | I,O | - | D |
| B27 | DDR1_DQ[24] | DDR | VDDQ | not shared | DDR1_DQ[24] | I,O | - | D |
| D27 | DDR1_DQ[25] | DDR | VDDQ | not shared | DDR1_DQ[25] | I,O | - | D |
| B23 | DDR1_DQ[26] | DDR | VDDQ | not shared | DDR1_DQ[26] | I,O | - | D |
| D23 | DDR1_DQ[27] | DDR | VDDQ | not shared | DDR1_DQ[27] | I,O | - | D |
| A28 | DDR1_DQ[28] | DDR | VDDQ | not shared | DDR1_DQ[28] | I,O | - | D |
| C28 | DDR1_DQ[29] | DDR | VDDQ | not shared | DDR1_DQ[29] | I,O | - | D |
| A24 | DDR1_DQ[30] | DDR | VDDQ | not shared | DDR1_DQ[30] | I,O | - | D |
| C24 | DDR1_DQ[31] | DDR | VDDQ | not shared | DDR1_DQ[31] | I,O | - | D |
| AP11 | DDR1_DQ[32] | DDR | VDDQ | not shared | DDR1_DQ[32] | I,O | - | D |
| AP7 | DDR1_DQ[33] | DDR | VDDQ | not shared | DDR1_DQ[33] | I,O | - | D |
| AY7 | DDR1_DQ[34] | DDR | VDDQ | not shared | DDR1_DQ[34] | I,O | - | D |
| AY11 | DDR1_DQ[35] | DDR | VDDQ | not shared | DDR1_DQ[35] | I,O | - | D |
| AN9 | DDR1_DQ[36] | DDR | VDDQ | not shared | DDR1_DQ[36] | I,O | - | D |
| AR9 | DDR1_DQ[37] | DDR | VDDQ | not shared | DDR1_DQ[37] | I,O | - | D |
| AW9 | DDR1_DQ[38] | DDR | VDDQ | not shared | DDR1_DQ[38] | I,O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 8 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| AV11 | DDR1_DQ[39] | DDR | VDDQ | not shared | DDR1_DQ[39] | I,O | - | D |
| AT15 | DDR1_DQ[40] | DDR | VDDQ | not shared | DDR1_DQ[40] | I,O | - | D |
| AU17 | DDR1_DQ[41] | DDR | VDDQ | not shared | DDR1_DQ[41] | I,O | - | D |
| BD17 | DDR1_DQ[42] | DDR | VDDQ | not shared | DDR1_DQ[42] | I,O | - | D |
| BB15 | DDR1_DQ[43] | DDR | VDDQ | not shared | DDR1_DQ[43] | I,O | - | D |
| AR17 | DDR1_DQ[44] | DDR | VDDQ | not shared | DDR1_DQ[44] | I,O | - | D |
| AP15 | DDR1_DQ[45] | DDR | VDDQ | not shared | DDR1_DQ[45] | I,O | - | D |
| BA17 | DDR1_DQ[46] | DDR | VDDQ | not shared | DDR1_DQ[46] | I,O | - | D |
| AY15 | DDR1_DQ[47] | DDR | VDDQ | not shared | DDR1_DQ[47] | I,O | - | D |
| BJ7 | DDR1_DQ[48] | DDR | VDDQ | not shared | DDR1_DQ[48] | I,O | - | D |
| BH9 | DDR1_DQ[49] | DDR | VDDQ | not shared | DDR1_DQ[49] | I,O | - | D |
| BL11 | DDR1_DQ[50] | DDR | VDDQ | not shared | DDR1_DQ[50] | I,O | - | D |
| BR9 | DDR1_DQ[51] | DDR | VDDQ | not shared | DDR1_DQ[51] | I,O | - | D |
| BF9 | DDR1_DQ[52] | DDR | VDDQ | not shared | DDR1_DQ[52] | I,O | - | D |
| BG7 | DDR1_DQ[53] | DDR | VDDQ | not shared | DDR1_DQ[53] | I,O | - | D |
| BN7 | DDR1_DQ[54] | DDR | VDDQ | not shared | DDR1_DQ[54] | I,O | - | D |
| BM9 | DDR1_DQ[55] | DDR | VDDQ | not shared | DDR1_DQ[55] | I,O | - | D |
| BH17 | DDR1_DQ[56] | DDR | VDDQ | not shared | DDR1_DQ[56] | I,O | - | D |
| BK17 | DDR1_DQ[57] | DDR | VDDQ | not shared | DDR1_DQ[57] | I,O | - | D |
| BM17 | DDR1_DQ[58] | DDR | VDDQ | not shared | DDR1_DQ[58] | I,O | - | D |
| BR17 | DDR1_DQ[59] | DDR | VDDQ | not shared | DDR1_DQ[59] | I,O | - | D |
| BG15 | DDR1_DQ[60] | DDR | VDDQ | not shared | DDR1_DQ[60] | I,O | - | D |
| BH13 | DDR1_DQ[61] | DDR | VDDQ | not shared | DDR1_DQ[61] | I,O | - | D |
| BR13 | DDR1_DQ[62] | DDR | VDDQ | not shared | DDR1_DQ[62] | I,O | - | D |
| BN15 | DDR1_DQ[63] | DDR | VDDQ | not shared | DDR1_DQ[63] | I,O | - | D |
| A32 | DDR1_DQS_DN[0] | DDR | VDDQ | not shared | DDR1_DQS_DN[0] | I,O, Diff | - | D |
| P22 | DDR1_DQS_DN[1] | DDR | VDDQ | not shared | DDR1_DQS_DN[1] | I,O, Diff | - | D |
| G22 | DDR1_DQS_DN[2] | DDR | VDDQ | not shared | DDR1_DQS_DN[2] | I,O, Diff | - | D |
| B25 | DDR1_DQS_DN[3] | DDR | VDDQ | not shared | DDR1_DQS_DN[3] | I,O, Diff | - | D |
| AT7 | DDR1_DQS_DN[4] | DDR | VDDQ | not shared | DDR1_DQS_DN[4] | I,O, Diff | - | D |
| AW17 | DDR1_DQS_DN[5] | DDR | VDDQ | not shared | DDR1_DQS_DN[5] | I,O, Diff | - | D |
| BL7 | DDR1_DQS_DN[6] | DDR | VDDQ | not shared | DDR1_DQS_DN[6] | I,O, Diff | - | D |
| BM13 | DDR1_DQS_DN[7] | DDR | VDDQ | not shared | DDR1_DQS_DN[7] | I,O, Diff | - | D |
| A18 | DDR1_DQS_DN[8] | DDR | VDDQ | not shared | DDR1_DQS_DN[8] | I,O, Diff | - | D |
| D33 | DDR1_DQS_DN[9] | DDR | VDDQ | not shared | DDR1_DQS_DN[9] | I,O, Diff | - | D |
| U24 | DDR1_DQS_DN[10] | DDR | VDDQ | not shared | DDR1_DQS_DN[10] | I,O, Diff | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 9 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| K24 | DDR1_DQS_DN[11] | DDR | VDDQ | not shared | DDR1_DQS_DN[11] | I,O, Diff | - | D |
| C26 | DDR1_DQS_DN[12] | DDR | VDDQ | not shared | DDR1_DQS_DN[12] | I,O, Diff | - | D |
| AT11 | DDR1_DQS_DN[13] | DDR | VDDQ | not shared | DDR1_DQS_DN[13] | I,O, Diff | - | D |
| AU13 | DDR1_DQS_DN[14] | DDR | VDDQ | not shared | DDR1_DQS_DN[14] | I,O, Diff | - | D |
| BE11 | DDR1_DQS_DN[15] | DDR | VDDQ | not shared | DDR1_DQS_DN[15] | I,O, Diff | - | D |
| BJ15 | DDR1_DQS_DN[16] | DDR | VDDQ | not shared | DDR1_DQS_DN[16] | I,O, Diff | - | D |
| D19 | DDR1_DQS_DN[17] | DDR | VDDQ | not shared | DDR1_DQS_DN[17] | I,O, Diff | - | D |
| C32 | DDR1_DQS_DP[0] | DDR | VDDQ | not shared | DDR1_DQS_DP[0] | I,O, Diff | - | D |
| T22 | DDR1_DQS_DP[1] | DDR | VDDQ | not shared | DDR1_DQS_DP[1] | I,O, Diff | - | D |
| J22 | DDR1_DQS_DP[2] | DDR | VDDQ | not shared | DDR1_DQS_DP[2] | I,O, Diff | - | D |
| D25 | DDR1_DQS_DP[3] | DDR | VDDQ | not shared | DDR1_DQS_DP[3] | I,O, Diff | - | D |
| AV7 | DDR1_DQS_DP[4] | DDR | VDDQ | not shared | DDR1_DQS_DP[4] | I,O, Diff | - | D |
| AV15 | DDR1_DQS_DP[5] | DDR | VDDQ | not shared | DDR1_DQS_DP[5] | I,O, Diff | - | D |
| BK9 | DDR1_DQS_DP[6] | DDR | VDDQ | not shared | DDR1_DQS_DP[6] | I,O, Diff | - | D |
| BL15 | DDR1_DQS_DP[7] | DDR | VDDQ | not shared | DDR1_DQS_DP[7] | I,O, Diff | - | D |
| C18 | DDR1_DQS_DP[8] | DDR | VDDQ | not shared | DDR1_DQS_DP[8] | I,O, Diff | - | D |
| B33 | DDR1_DQS_DP[9] | DDR | VDDQ | not shared | DDR1_DQS_DP[9] | I,O, Diff | - | D |
| R24 | DDR1_DQS_DP[10] | DDR | VDDQ | not shared | DDR1_DQS_DP[10] | I,O, Diff | - | D |
| H24 | DDR1_DQS_DP[11] | DDR | VDDQ | not shared | DDR1_DQS_DP[11] | I,O, Diff | - | D |
| A26 | DDR1_DQS_DP[12] | DDR | VDDQ | not shared | DDR1_DQS_DP[12] | I,O, Diff | - | D |
| AU9 | DDR1_DQS_DP[13] | DDR | VDDQ | not shared | DDR1_DQS_DP[13] | I,O, Diff | - | D |
| AW13 | DDR1_DQS_DP[14] | DDR | VDDQ | not shared | DDR1_DQS_DP[14] | I,O, Diff | - | D |
| BG11 | DDR1_DQS_DP[15] | DDR | VDDQ | not shared | DDR1_DQS_DP[15] | I,O, Diff | - | D |
| BK13 | DDR1_DQS_DP[16] | DDR | VDDQ | not shared | DDR1_DQS_DP[16] | I,O, Diff | - | D |
| B19 | DDR1_DQS_DP[17] | DDR | VDDQ | not shared | DDR1_DQS_DP[17] | I,O, Diff | - | D |
| A20 | DDR1_ECC[0] | DDR | VDDQ | not shared | DDR1_ECC[0] | I,O | - | D |
| C20 | DDR1_ECC[1] | DDR | VDDQ | not shared | DDR1_ECC[1] | I,O | - | D |
| A15 | DDR1_ECC[2] | DDR | VDDQ | not shared | DDR1_ECC[2] | I,O | - | D |
| C15 | DDR1_ECC[3] | DDR | VDDQ | not shared | DDR1_ECC[3] | I,O | - | D |
| B21 | DDR1_ECC[4] | DDR | VDDQ | not shared | DDR1_ECC[4] | I,O | - | D |
| D21 | DDR1_ECC[5] | DDR | VDDQ | not shared | DDR1_ECC[5] | I,O | - | D |
| B16 | DDR1_ECC[6] | DDR | VDDQ | not shared | DDR1_ECC[6] | I,O | - | D |
| D16 | DDR1_ECC[7] | DDR | VDDQ | not shared | DDR1_ECC[7] | I,O | - | D |
| AE22 | DDR1_MA[0] | DDR | VDDQ | not shared | DDR1_MA[0] | O | - | D |
| U9 | DDR1_MA[1] | DDR | VDDQ | not shared | DDR1_MA[1] | O | - | D |
| T7 | DDR1_MA[2] | DDR | VDDQ | not shared | DDR1_MA[2] | O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 10 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| P7 | DDR1_MA[3] | DDR | VDDQ | not shared | DDR1_MA[3] | O | - | D |
| R13 | DDR1_MA[4] | DDR | VDDQ | not shared | DDR1_MA[4] | O | - | D |
| N13 | DDR1_MA[5] | DDR | VDDQ | not shared | DDR1_MA[5] | O | - | D |
| M7 | DDR1_MA[6] | DDR | VDDQ | not shared | DDR1_MA[6] | O | - | D |
| N9 | DDR1_MA[7] | DDR | VDDQ | not shared | DDR1_MA[7] | O | - | D |
| R9 | DDR1_MA[8] | DDR | VDDQ | not shared | DDR1_MA[8] | O | - | D |
| J11 | DDR1_MA[9] | DDR | VDDQ | not shared | DDR1_MA[9] | O | - | D |
| AF20 | DDR1_MA[10] | DDR | VDDQ | not shared | DDR1_MA[10] | O | - | D |
| J7 | DDR1_MA[11] | DDR | VDDQ | not shared | DDR1_MA[11] | O | - | D |
| G11 | DDR1_MA[12] | DDR | VDDQ | not shared | DDR1_MA[12] | O | - | D |
| AW20 | DDR1_MA[13] | DDR | VDDQ | not shared | DDR1_MA[13] | O | - | D |
| AR20 | DDR1_MA14_WE_N | DDR | VDDQ | not shared | DDR1_MA14_WE_N | O | - | D |
| AT22 | DDR1_MA15_CAS_N | DDR | VDDQ | not shared | DDR1_MA15_CAS_N | O | - | D |
| AC22 | DDR1_MA16_RAS_N | DDR | VDDQ | not shared | DDR1_MA16_RAS_N | O | - | D |
| AG15 | DDR1_MA17 | DDR | VDDQ | not shared | DDR1_MA17 | O | - | D |
| AU20 | DDR1_ODT[0] | DDR | VDDQ | not shared | DDR1_ODT[0] | O | - | D |
| AY22 | DDR1_ODT[1] | DDR | VDDQ | not shared | DDR1_ODT[1] | O | - | D |
| AE15 | DDR1_ODT[2] | DDR | VDDQ | not shared | DDR1_ODT[2] | O | - | D |
| AK17 | DDR1_ODT[3] | DDR | VDDQ | not shared | DDR1_ODT[3] | O | - | D |
| V7 | DDR1_PAR | DDR | VDDQ | not shared | DDR1_PAR | O | - | D |
| BN68 | DFX_PORT[0] | CMOS-LV | V1P05 | Default | DFX_PORT[0] | O | - | D |
| | | | | Func 0 GPIO | GPIO[55] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[0] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | I,O | - | |
| | | | | Func 4 | Reserved | I,O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| BN70 | DFX_PORT[1] | CMOS-LV | V1P05 | Default | DFX_PORT[1] | O | - | D |
| | | | | Func 0 GPIO | GPIO[56] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[1] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | I,O | - | |
| | | | | Func 4 | Reserved | I,O | - | |
| | | | | Func 5 | Reserved | O | - | |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 11 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BE63 | DFX_PORT[2] | CMOS-LV | V1P05 | Default | DFX_PORT[2] | O | - | D |
| | | | | Func 0 GPIO | GPIO[57] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[2] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | I,O | - | |
| | | | | Func 4 | Reserved | I,O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BP69 | DFX_PORT[3] | CMOS-LV | V1P05 | Default | DFX_PORT[3] | O | - | D |
| | | | | Func 0 GPIO | GPIO[58] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[3] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | I,O | - | |
| | | | | Func 4 | Reserved | I,O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BW68 | DFX_PORT[4] | CMOS-LV | V1P05 | Default | DFX_PORT[4] | O | - | D |
| | | | | Func 0 GPIO | GPIO[59] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[4] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BH69 | DFX_PORT[5] | CMOS-LV | V1P05 | Default | DFX_PORT[5] | O | - | D |
| | | | | Func 0 GPIO | GPIO[60] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[5] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BG63 | DFX_PORT[6] | CMOS-LV | V1P05 | Default | DFX_PORT[6] | O | - | D |
| | | | | Func 0 GPIO | GPIO[61] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[6] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 12 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BT69 | DFX_PORT[7] | CMOS-LV | V1P05 | Default | DFX_PORT[7] | O | - | D |
| | | | | Func 0 GPIO | GPIO[62] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[7] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BW67 | DFX_PORT[8] | CMOS-LV | V1P05 | Default | DFX_PORT[8] | O | - | D |
| | | | | Func 0 GPIO | GPIO[63] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[8] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| AY52 | DFX_PORT[9] | CMOS-LV | V1P05 | Default | DFX_PORT[9] | O | - | D |
| | | | | Func 0 GPIO | GPIO[134] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[9] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| BR42 | DFX_PORT[10] | CMOS-LV | V1P05 | Default | DFX_PORT[10] | O | - | D |
| | | | | Func 0 GPIO | GPIO[135] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[10] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 13 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BK42 | DFX_PORT[11] | CMOS-LV | V1P05 | Default | DFX_PORT[11] | O | - | D |
| | | | | Func 0 GPIO | GPIO[136] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[11] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| BW41 | DFX_PORT[12] | CMOS-LV | V1P05 | Default | DFX_PORT[12] | O | - | D |
| | | | | Func 0 GPIO | GPIO[137] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[12] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| BE48 | DFX_PORT[13] | CMOS-LV | V1P05 | Default | DFX_PORT[13] | O | - | D |
| | | | | Func 0 GPIO | GPIO[138] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[13] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| | | | | Func 6 | Reserved | O | - | |
| CA41 | DFX_PORT[14] | CMOS-LV | V1P05 | Default | DFX_PORT[14] | O | - | D |
| | | | | Func 0 GPIO | GPIO[139] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[14] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 3 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |
| BV40 | DFX_PORT[15] | CMOS-LV | V1P05 | Default | DFX_PORT[15] | O | - | D |
| | | | | Func 0 GPIO | GPIO[140] | SW | SW | D |
| | | | | Func 1 | DFX_PORT[15] | O | - | D |
| | | | | Func 2 | Reserved | O | - | |
| | | | | Func 4 | Reserved | O | - | |
| | | | | Func 5 | Reserved | O | - | |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 14 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| AY56 | DFX_PORT_CLK[0] | CMOS-LV | V1P05 | Default | DFX_PORT_CLK[0] | O | - | D |
| | | | | Func 0 GPIO | GPIO[53] | SW | SW | D |
| | | | | Func 1 | DFX_PORT_CLK[0] | O | - | D |
| BB70 | DFX_PORT_CLK[1] | CMOS-LV | V1P05 | Default | DFX_PORT_CLK[1] | O | - | D |
| | | | | Func 0 GPIO | GPIO[54] | SW | SW | D |
| | | | | Func 1 | DFX_PORT_CLK[1] | O | - | D |
| | | | | Func 2 | Reserved | I | - | |
| BK46 | EMMC_CLK | CMOS-MV | V1P80 | Default | GPIO[125] | I | - | D |
| | | | | Func 0 GPIO | GPIO[125] | SW | SW | D |
| | | | | Func 1 | EMMC_CLK | I,O | - | D |
| BN48 | EMMC_CMD | CMOS-MV | V1P80 | Default | GPIO[123] | I | - | D |
| | | | | Func 0 GPIO | GPIO[123] | SW | SW | D |
| | | | | Func 1 | EMMC_CMD | I,O | - | D |
| BW45 | EMMC_D[0] | CMOS-MV | V1P80 | Default | GPIO[126] | I | - | D |
| | | | | Func 0 GPIO | GPIO[126] | SW | SW | D |
| | | | | Func 1 | EMMC_D[0] | I,O | - | D |
| BV44 | EMMC_D[1] | CMOS-MV | V1P80 | Default | GPIO[127] | I | - | D |
| | | | | Func 0 GPIO | GPIO[127] | SW | SW | D |
| | | | | Func 1 | EMMC_D[1] | I,O | - | D |
| BY44 | EMMC_D[2] | CMOS-MV | V1P80 | Default | GPIO[128] | I | - | D |
| | | | | Func 0 GPIO | GPIO[128] | SW | SW | D |
| | | | | Func 1 | EMMC_D[2] | I,O | - | D |
| CA45 | EMMC_D[3] | CMOS-MV | V1P80 | Default | GPIO[129] | I | - | D |
| | | | | Func 0 GPIO | GPIO[129] | SW | SW | D |
| | | | | Func 1 | EMMC_D[3] | I,O | - | D |
| BU43 | EMMC_D[4] | CMOS-MV | V1P80 | Default | GPIO[130] | I | - | D |
| | | | | Func 0 GPIO | GPIO[130] | SW | SW | D |
| | | | | Func 1 | EMMC_D[4] | I,O | - | D |
| CA43 | EMMC_D[5] | CMOS-MV | V1P80 | Default | GPIO[131] | I | - | D |
| | | | | Func 0 GPIO | GPIO[131] | SW | SW | D |
| | | | | Func 1 | EMMC_D[5] | I,O | - | D |
| BY42 | EMMC_D[6] | CMOS-MV | V1P80 | Default | GPIO[132] | I | - | D |
| | | | | Func 0 GPIO | GPIO[132] | SW | SW | D |
| | | | | Func 1 | EMMC_D[6] | I,O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 15 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BV42 | EMMC_D[7] | CMOS-MV | V1P80 | Default | GPIO[133] | I | - | D |
| | | | | Func 0 GPIO | GPIO[133] | SW | SW | D |
| | | | | Func 1 | EMMC_D[7] | I,O | - | D |
| BE40 | EMMC_IRCOMP | Analog | V1P80 | not shared | EMMC_IRCOMP | I,O | - | D |
| BL44 | EMMC_STROBE | CMOS-MV | V1P80 | Default | GPIO[124] | I | - | D |
| | | | | Func 0 GPIO | GPIO[124] | SW | SW | D |
| | | | | Func 1 | EMMC_STROBE | I | - | D |
| BL52 | ERROR_N[0] | CMOS-HV | V3P30 | Strap Sampling | LPC Decode Select | I | 20K PD | D |
| | | | | Default | ERROR_N[0] | O | - | D |
| | | | | Func 0 GPIO | GPIO[107] | SW | SW | D |
| | | | | Func 1 | ERROR_N[0] | O | - | D |
| BR58 | ERROR_N[1] | CMOS-HV | V3P30 | Strap Sampling | Boot BIOS Strap (BBS) | I | 20K PD | D |
| | | | | Default | ERROR_N[1] | O | - | D |
| | | | | Func 0 GPIO | GPIO[106] | SW | SW | D |
| | | | | Func 1 | ERROR_N[1] | O | - | D |
| CA50 | ERROR_N[2] | CMOS-HV | V3P30 | Strap Sampling | Intel Reserved | I | 20K PD | D |
| | | | | Default | ERROR_N[2] | O | - | D |
| | | | | Func 0 GPIO | GPIO[105] | SW | SW | D |
| | | | | Func 1 | ERROR_N[2] | O | - | D |
| BK61 | ESPI_ALRT_N[0] | CMOS-HV | V3P30 | Default | LPC_CLKRUN_N | | | |
| | | | | Func 0 GPIO | GPIO[71] | SW | SW | D |
| | | | | Func 2 | LPC_CLKRUN_N | I,O-Tri | - | D |
| BY62 | ESPI_ALRT_N[1] | CMOS-HV | V3P30 | Default | ESPI_ALRT_N[1] | I | - | D |
| | | | | Func 0 GPIO | GPIO[11] | SW | SW | D |
| | | | | Func 1 | ESPI_ALRT_N[1] | I | - | D |
| | | | | Func 2 | LPC_SERIRQ | I,O-Tri | - | D |
| BY64 | ESPI_CLK | CMOS-HV | V3P30 | Default | LPC_CLKOUT[0] | | | |
| | | | | Func 0 GPIO | GPIO[69] | SW | SW | D |
| | | | | Func 2 | LPC_CLKOUT[0] | O | 20K PD | D |
| BJ72 | ESPI_CS_N[0] | CMOS-HV | V3P30 | Default | LPC_FRAME_N | | | |
| | | | | Func 0 GPIO | GPIO[68] | SW | SW | D |
| | | | | Func 2 | LPC_FRAME_N | O | 20K PU | D |
| CA63 | ESPI_CS_N[1] | CMOS-HV | V3P30 | Default | ESPI_CS_N[1] | O-Tri | - | D |
| | | | | Func 0 GPIO | GPIO[10] | SW | SW | D |
| | | | | Func 1 | ESPI_CS_N[1] | O-Tri | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 16 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BD58 | ESPI_IO[0] | CMOS-HV | V3P30 | Default | LPC_AD[0] | | | |
| | | | | Func 0 GPIO | GPIO[64] | SW | SW | D |
| | | | | Func 2 | LPC_AD[0] | I,O-Tri | 20K PU | D |
| BR67 | ESPI_IO[1] | CMOS-HV | V3P30 | Default | LPC_AD[1] | | | |
| | | | | Func 0 GPIO | GPIO[65] | SW | SW | D |
| | | | | Func 2 | LPC_AD[1] | I,O-Tri | 20K PU | D |
| BV64 | ESPI_IO[2] | CMOS-HV | V3P30 | Default | LPC_AD[2] | | | |
| | | | | Func 0 GPIO | GPIO[66] | SW | SW | D |
| | | | | Func 2 | LPC_AD[2] | I,O-Tri | 20K PU | D |
| BH65 | ESPI_IO[3] | CMOS-HV | V3P30 | Default | LPC_AD[3] | | | |
| | | | | Func 0 GPIO | GPIO[67] | SW | SW | D |
| | | | | Func 2 | LPC_AD[3] | I,O-Tri | 20K PU | D |
| BD50 | ESPI_IRCOMP | Analog | V3P30 | not shared | ESPI_IRCOMP | I,O | - | D |
| BV66 | ESPI_RST_N | CMOS-HV | V3P30 | Default | LPC_CLKOUT[1] | | | |
| | | | | Func 0 GPIO | GPIO[70] | SW | SW | D |
| | | | | Func 2 | LPC_CLKOUT[1] | O | 20K PD | D |
| BG48 | FLEX_CLK_SE[0] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PD | D |
| | | | | Default | FLEX_CLK_SE[0] | O | - | D |
| | | | | Func 0 GPIO | GPIO[121] | SW | SW | D |
| | | | | Func 1 | FLEX_CLK_SE[0] | O | - | D |
| CA48 | FLEX_CLK_SE[1] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[122] | O | - | D |
| | | | | Func 0 GPIO | GPIO[122] | SW | SW | D |
| | | | | Func 1 | FLEX_CLK_SE[1] | O | - | D |
| BF70 | GPIO[0] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[0] | O | - | D |
| | | | | Func 0 GPIO | GPIO[0] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O | - | |
| | | | | Func 2 | FAN_TACH[2] | I | - | D |
| | | | | Func 3 | ADR_COMPLETE | I,O | - | D |
| BK69 | GPIO[1] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[1] | O | - | D |
| | | | | Func 0 GPIO | GPIO[1] | SW | SW | D |
| | | | | Func 1 | LAN_MDC | O | - | D |
| | | | | Func 2 | Reserved | I,O | - | |
| | | | | Func 3 | FAN_TACH[0] | I | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 17 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BK71 | GPIO[2] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[2] | O | - | D |
| | | | | Func 0 GPIO | GPIO[2] | SW | SW | D |
| | | | | Func 1 | LAN_MDIO | I,O-OD | - | D |
| | | | | Func 2 | Reserved | I,O | - | |
| | | | | Func 3 | FAN_TACH[1] | I | - | D |
| BN44 | GPIO[3] | CMOS-MV | V1P80 | Strap Sampling | Top Swap Override | I | 20K PD | D |
| | | | | Default | GPIO[3] | O | - | D |
| | | | | Func 0 GPIO | GPIO[3] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O | - | |
| BH50 | GPIO[4] | CMOS-HV | V3P30 | Default | GPIO[4] | O | - | D |
| | | | | Func 0 GPIO | GPIO[4] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O | - | |
| | | | | Func 2 | FAN_PWM[0] | O-OD | - | D |
| | | | | Func 3 | LAN1_PORT1_LED[1] | O-OD | - | D |
| BL48 | GPIO[5] | CMOS-HV | V3P30 | Default | GPIO[5] | I | - | D |
| | | | | Func 0 GPIO | GPIO[5] | SW | SW | D |
| | | | | Func 1 | SMB_IE_SMT0_CLK | I,O-OD | - | D |
| BE52 | GPIO[6] | CMOS-HV | V3P30 | Default | GPIO[6] | I | - | D |
| | | | | Func 0 GPIO | GPIO[6] | SW | SW | D |
| | | | | Func 1 | SMB_IE_SMT0_DATA | I,O-OD | - | D |
| BH46 | GPIO[7] | CMOS-HV | V3P30 | Default | GPIO[7] | I | - | D |
| | | | | Func 0 GPIO | GPIO[7] | SW | SW | D |
| | | | | Func 1 | SMB_IE_SMT0_ALRT_N | I,O-OD | - | D |
| BY55 | GPIO[8] | CMOS-HV | V3P30 | Default | GPIO[8] | I | - | D |
| | | | | Func 0 GPIO | GPIO[8] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O | - | |
| | | | | Func 2 | FAN_PWM[1] | O-OD | - | D |
| | | | | Func 3 | SMB_ME_SMT1_DATA | I,O-OD | - | D |
| CA54 | GPIO[9] | CMOS-HV | V3P30 | Default | GPIO[9] | I | - | D |
| | | | | Func 0 GPIO | GPIO[9] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O | - | |
| | | | | Func 2 | FAN_PWM[2] | O-OD | - | D |
| | | | | Func 3 | SMB_ME_SMT1_CLK | I,O-OD | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 18 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BN56 | GPIO[12] | CMOS-HV | V3P30 | Strap Sampling | No TCO Reboot Enable | I | 20K PD | D |
| | | | | Default | GPIO[12] | O | - | D |
| | | | | Func 0 GPIO | GPIO[12] | SW | SW | D |
| | | | | Func 1 | SPI_TPM_CS_N | I,O | - | D |
| BM38 | GPIO[150] | CMOS-LV | V1P05 | Default | GPIO[150] | I | - | D |
| | | | | Func 0 GPIO | GPIO[150] | SW | SW | D |
| | | | | Func 1 | Reserved | I | - | |
| BW37 | GPIO[151] | CMOS-LV | V1P05 | Default | GPIO[151] | I | - | D |
| | | | | Func 0 GPIO | GPIO[151] | SW | SW | D |
| | | | | Func 1 | Reserved | I | - | |
| CA37 | GPIO[152] | CMOS-LV | V1P05 | Default | GPIO[152] | I | - | D |
| | | | | Func 0 GPIO | GPIO[152] | SW | SW | D |
| | | | | Func 1 | Reserved | I,O-OD | - | |
| BD42 | HSHV_IRCOMP | HSIO | V3P30 | not shared | HSHV_IRCOMP | I,O | - | D |
| AC56 | HSIO_COMPREF_DN | HSIO | V1P05 | not shared | HSIO_COMPREF_DN | O, Diff | - | D |
| AD54 | HSIO_COMPREF_DP | HSIO | V1P05 | not shared | HSIO_COMPREF_DP | O, Diff | - | D |
| C52 | HSIO_RX_DN[0] | HSIO | V1P05 | not shared | HSIO_RX_DN[0] | I, Diff | - | D |
| C54 | HSIO_RX_DN[1] | HSIO | V1P05 | not shared | HSIO_RX_DN[1] | I, Diff | - | D |
| C57 | HSIO_RX_DN[2] | HSIO | V1P05 | not shared | HSIO_RX_DN[2] | I, Diff | - | D |
| C59 | HSIO_RX_DN[3] | HSIO | V1P05 | not shared | HSIO_RX_DN[3] | I, Diff | - | D |
| C61 | HSIO_RX_DN[4] | HSIO | V1P05 | not shared | HSIO_RX_DN[4] | I, Diff | - | D |
| C63 | HSIO_RX_DN[5] | HSIO | V1P05 | not shared | HSIO_RX_DN[5] | I, Diff | - | D |
| C65 | HSIO_RX_DN[6] | HSIO | V1P05 | not shared | HSIO_RX_DN[6] | I, Diff | - | D |
| F69 | HSIO_RX_DN[7] | HSIO | V1P05 | not shared | HSIO_RX_DN[7] | I, Diff | - | D |
| H69 | HSIO_RX_DN[8] | HSIO | V1P05 | not shared | HSIO_RX_DN[8] | I, Diff | - | D |
| L70 | HSIO_RX_DN[9] | HSIO | V1P05 | not shared | HSIO_RX_DN[9] | I, Diff | - | D |
| N70 | HSIO_RX_DN[10] | HSIO | V1P05 | not shared | HSIO_RX_DN[10] | I, Diff | - | D |
| R70 | HSIO_RX_DN[11] | HSIO | V1P05 | not shared | HSIO_RX_DN[11] | I, Diff | - | D |
| V70 | HSIO_RX_DN[12] | HSIO | V1P05 | not shared | HSIO_RX_DN[12] | I, Diff | - | D |
| Y70 | HSIO_RX_DN[13] | HSIO | V1P05 | not shared | HSIO_RX_DN[13] | I, Diff | - | D |
| AB70 | HSIO_RX_DN[14] | HSIO | V1P05 | not shared | HSIO_RX_DN[14] | I, Diff | - | D |
| AD70 | HSIO_RX_DN[15] | HSIO | V1P05 | not shared | HSIO_RX_DN[15] | I, Diff | - | D |
| AF70 | HSIO_RX_DN[16] | HSIO | V1P05 | not shared | HSIO_RX_DN[16] | I, Diff | - | D |
| AJ70 | HSIO_RX_DN[17] | HSIO | V1P05 | not shared | HSIO_RX_DN[17] | I, Diff | - | D |
| AL70 | HSIO_RX_DN[18] | HSIO | V1P05 | not shared | HSIO_RX_DN[18] | I, Diff | - | D |
| AN70 | HSIO_RX_DN[19] | HSIO | V1P05 | not shared | HSIO_RX_DN[19] | I, Diff | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 19 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| A52 | HSIO_RX_DP[0] | HSIO | V1P05 | not shared | HSIO_RX_DP[0] | I, Diff | - | D |
| A54 | HSIO_RX_DP[1] | HSIO | V1P05 | not shared | HSIO_RX_DP[1] | I, Diff | - | D |
| A57 | HSIO_RX_DP[2] | HSIO | V1P05 | not shared | HSIO_RX_DP[2] | I, Diff | - | D |
| A59 | HSIO_RX_DP[3] | HSIO | V1P05 | not shared | HSIO_RX_DP[3] | I, Diff | - | D |
| A61 | HSIO_RX_DP[4] | HSIO | V1P05 | not shared | HSIO_RX_DP[4] | I, Diff | - | D |
| A63 | HSIO_RX_DP[5] | HSIO | V1P05 | not shared | HSIO_RX_DP[5] | I, Diff | - | D |
| C67 | HSIO_RX_DP[6] | HSIO | V1P05 | not shared | HSIO_RX_DP[6] | I, Diff | - | D |
| G67 | HSIO_RX_DP[7] | HSIO | V1P05 | not shared | HSIO_RX_DP[7] | I, Diff | - | D |
| J70 | HSIO_RX_DP[8] | HSIO | V1P05 | not shared | HSIO_RX_DP[8] | I, Diff | - | D |
| L72 | HSIO_RX_DP[9] | HSIO | V1P05 | not shared | HSIO_RX_DP[9] | I, Diff | - | D |
| N72 | HSIO_RX_DP[10] | HSIO | V1P05 | not shared | HSIO_RX_DP[10] | I, Diff | - | D |
| R72 | HSIO_RX_DP[11] | HSIO | V1P05 | not shared | HSIO_RX_DP[11] | I, Diff | - | D |
| V72 | HSIO_RX_DP[12] | HSIO | V1P05 | not shared | HSIO_RX_DP[12] | I, Diff | - | D |
| Y72 | HSIO_RX_DP[13] | HSIO | V1P05 | not shared | HSIO_RX_DP[13] | I, Diff | - | D |
| AB72 | HSIO_RX_DP[14] | HSIO | V1P05 | not shared | HSIO_RX_DP[14] | I, Diff | - | D |
| AD72 | HSIO_RX_DP[15] | HSIO | V1P05 | not shared | HSIO_RX_DP[15] | I, Diff | - | D |
| AF72 | HSIO_RX_DP[16] | HSIO | V1P05 | not shared | HSIO_RX_DP[16] | I, Diff | - | D |
| AJ72 | HSIO_RX_DP[17] | HSIO | V1P05 | not shared | HSIO_RX_DP[17] | I, Diff | - | D |
| AL72 | HSIO_RX_DP[18] | HSIO | V1P05 | not shared | HSIO_RX_DP[18] | I, Diff | - | D |
| AN72 | HSIO_RX_DP[19] | HSIO | V1P05 | not shared | HSIO_RX_DP[19] | I, Diff | - | D |
| G48 | HSIO_TX_DN[0] | HSIO | V1P05 | not shared | HSIO_TX_DN[0] | O, Diff | - | D |
| K50 | HSIO_TX_DN[1] | HSIO | V1P05 | not shared | HSIO_TX_DN[1] | O, Diff | - | D |
| J52 | HSIO_TX_DN[2] | HSIO | V1P05 | not shared | HSIO_TX_DN[2] | O, Diff | - | D |
| H54 | HSIO_TX_DN[3] | HSIO | V1P05 | not shared | HSIO_TX_DN[3] | O, Diff | - | D |
| G56 | HSIO_TX_DN[4] | HSIO | V1P05 | not shared | HSIO_TX_DN[4] | O, Diff | - | D |
| K58 | HSIO_TX_DN[5] | HSIO | V1P05 | not shared | HSIO_TX_DN[5] | O, Diff | - | D |
| J59 | HSIO_TX_DN[6] | HSIO | V1P05 | not shared | HSIO_TX_DN[6] | O, Diff | - | D |
| H61 | HSIO_TX_DN[7] | HSIO | V1P05 | not shared | HSIO_TX_DN[7] | O, Diff | - | D |
| J63 | HSIO_TX_DN[8] | HSIO | V1P05 | not shared | HSIO_TX_DN[8] | O, Diff | - | D |
| J66 | HSIO_TX_DN[9] | HSIO | V1P05 | not shared | HSIO_TX_DN[9] | O, Diff | - | D |
| M63 | HSIO_TX_DN[10] | HSIO | V1P05 | not shared | HSIO_TX_DN[10] | O, Diff | - | D |
| P66 | HSIO_TX_DN[11] | HSIO | V1P05 | not shared | HSIO_TX_DN[11] | O, Diff | - | D |
| T63 | HSIO_TX_DN[12] | HSIO | V1P05 | not shared | HSIO_TX_DN[12] | O, Diff | - | D |
| V66 | HSIO_TX_DN[13] | HSIO | V1P05 | not shared | HSIO_TX_DN[13] | O, Diff | - | D |
| Y63 | HSIO_TX_DN[14] | HSIO | V1P05 | not shared | HSIO_TX_DN[14] | O, Diff | - | D |
| AC66 | HSIO_TX_DN[15] | HSIO | V1P05 | not shared | HSIO_TX_DN[15] | O, Diff | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 20 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| AE63 | HSIO_TX_DN[16] | HSIO | V1P05 | not shared | HSIO_TX_DN[16] | O, Diff | - | D |
| AG66 | HSIO_TX_DN[17] | HSIO | V1P05 | not shared | HSIO_TX_DN[17] | O, Diff | - | D |
| AJ63 | HSIO_TX_DN[18] | HSIO | V1P05 | not shared | HSIO_TX_DN[18] | O, Diff | - | D |
| AL66 | HSIO_TX_DN[19] | HSIO | V1P05 | not shared | HSIO_TX_DN[19] | O, Diff | - | D |
| J48 | HSIO_TX_DP[0] | HSIO | V1P05 | not shared | HSIO_TX_DP[0] | O, Diff | - | D |
| M48 | HSIO_TX_DP[1] | HSIO | V1P05 | not shared | HSIO_TX_DP[1] | O, Diff | - | D |
| M52 | HSIO_TX_DP[2] | HSIO | V1P05 | not shared | HSIO_TX_DP[2] | O, Diff | - | D |
| G52 | HSIO_TX_DP[3] | HSIO | V1P05 | not shared | HSIO_TX_DP[3] | O, Diff | - | D |
| J56 | HSIO_TX_DP[4] | HSIO | V1P05 | not shared | HSIO_TX_DP[4] | O, Diff | - | D |
| M56 | HSIO_TX_DP[5] | HSIO | V1P05 | not shared | HSIO_TX_DP[5] | O, Diff | - | D |
| M59 | HSIO_TX_DP[6] | HSIO | V1P05 | not shared | HSIO_TX_DP[6] | O, Diff | - | D |
| G59 | HSIO_TX_DP[7] | HSIO | V1P05 | not shared | HSIO_TX_DP[7] | O, Diff | - | D |
| G63 | HSIO_TX_DP[8] | HSIO | V1P05 | not shared | HSIO_TX_DP[8] | O, Diff | - | D |
| H65 | HSIO_TX_DP[9] | HSIO | V1P05 | not shared | HSIO_TX_DP[9] | O, Diff | - | D |
| K65 | HSIO_TX_DP[10] | HSIO | V1P05 | not shared | HSIO_TX_DP[10] | O, Diff | - | D |
| N65 | HSIO_TX_DP[11] | HSIO | V1P05 | not shared | HSIO_TX_DP[11] | O, Diff | - | D |
| R65 | HSIO_TX_DP[12] | HSIO | V1P05 | not shared | HSIO_TX_DP[12] | O, Diff | - | D |
| U65 | HSIO_TX_DP[13] | HSIO | V1P05 | not shared | HSIO_TX_DP[13] | O, Diff | - | D |
| W65 | HSIO_TX_DP[14] | HSIO | V1P05 | not shared | HSIO_TX_DP[14] | O, Diff | - | D |
| AA65 | HSIO_TX_DP[15] | HSIO | V1P05 | not shared | HSIO_TX_DP[15] | O, Diff | - | D |
| AD65 | HSIO_TX_DP[16] | HSIO | V1P05 | not shared | HSIO_TX_DP[16] | O, Diff | - | D |
| AF65 | HSIO_TX_DP[17] | HSIO | V1P05 | not shared | HSIO_TX_DP[17] | O, Diff | - | D |
| AH65 | HSIO_TX_DP[18] | HSIO | V1P05 | not shared | HSIO_TX_DP[18] | O, Diff | - | D |
| AK65 | HSIO_TX_DP[19] | HSIO | V1P05 | not shared | HSIO_TX_DP[19] | O, Diff | - | D |
| W54 | ICLKRCOMP | ISCLK | V1P05 | not shared | ICLKRCOMP | I,O | - | D |
| BW50 | IERR_N | CMOS-HV | V3P30 | Strap Sampling | Flash Security Override | I | 20K PD | D |
| | | | | Default | IERR_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[108] | SW | SW | D |
| | | | | Func 1 | IERR_N | O | - | D |
| AJ59 | INTRUDER_N | RTC | VRTC | not shared | INTRUDER_N | I | - | D |
| AV59 | LAN0_PORT0_I2C_CLK | CMOS-HV | V3P30 | Default | LAN0_PORT0_I2C_CLK | I,O-OD | 5K PU | D |
| | | | | Func 0 GPIO | GPIO[24] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_I2C_CLK | I,O-OD | 5K PU | D |
| AV56 | LAN0_PORT0_I2C_DATA | CMOS-HV | V3P30 | Default | LAN0_PORT0_I2C_DATA | I,O-OD | 5K PU | D |
| | | | | Func 0 GPIO | GPIO[25] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_I2C_DATA | I,O-OD | 5K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 21 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BE69 | LAN0_PORT0_LED[0] | CMOS-HV | V3P30 | Default | LAN0_PORT0_LED[0] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[37] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_LED[0] | O-OD | - | D |
| BF72 | LAN0_PORT0_LED[1] | CMOS-HV | V3P30 | Default | LAN0_PORT0_LED[1] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[38] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_LED[1] | O-OD | - | D |
| U42 | LAN0_PORT0_RX_DN | KR | V1P05 | not shared | LAN0_PORT0_RX_DN | I, Diff | - | D |
| R42 | LAN0_PORT0_RX_DP | KR | V1P05 | not shared | LAN0_PORT0_RX_DP | I, Diff | - | D |
| AV66 | LAN0_PORT0_SDP[0] | CMOS-HV | V3P30 | Default | GPIO[14] | O | - | D |
| | | | | Func 0 GPIO | GPIO[14] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_SDP[0] | I,O | - | D |
| BA71 | LAN0_PORT0_SDP[1] | CMOS-HV | V3P30 | Default | GPIO[16] | O | - | D |
| | | | | Func 0 GPIO | GPIO[16] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_SDP[1] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT0_I2C_CLK | I,O-OD | - | D |
| AC59 | LAN0_PORT0_SDP[2] | CMOS-HV | V3P30 | Default | GPIO[18] | O | - | D |
| | | | | Func 0 GPIO | GPIO[18] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_SDP[2] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT0_SDP[0] | I,O | - | D |
| BA69 | LAN0_PORT0_SDP[3] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[20] | O | - | D |
| | | | | Func 0 GPIO | GPIO[20] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT0_SDP[3] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT1_I2C_CLK | I,O-OD | - | D |
| | | | | Func 3 | LAN1_PORT0_SDP[1] | I,O | - | D |
| G40 | LAN0_PORT0_TX_DN | KR | V1P05 | not shared | LAN0_PORT0_TX_DN | O, Diff | - | D |
| J40 | LAN0_PORT0_TX_DP | KR | V1P05 | not shared | LAN0_PORT0_TX_DP | O, Diff | - | D |
| BB66 | LAN0_PORT1_I2C_CLK | CMOS-HV | V3P30 | Default | LAN0_PORT1_I2C_CLK | I,O-OD | 5K PU | D |
| | | | | Func 0 GPIO | GPIO[26] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_I2C_CLK | I,O-OD | 5K PU | D |
| BC69 | LAN0_PORT1_I2C_DATA | CMOS-HV | V3P30 | Default | LAN0_PORT1_I2C_DATA | I,O-OD | 5K PU | D |
| | | | | Func 0 GPIO | GPIO[27] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_I2C_DATA | I,O-OD | 5K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 22 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BA65 | LAN0_PORT1_LED[0] | CMOS-HV | V3P30 | Default | LAN0_PORT1_LED[0] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[39] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_LED[0] | O-OD | - | D |
| | | | | Func 2 | LAN0_PORT0_LED[2] | O-OD | - | D |
| BG66 | LAN0_PORT1_LED[1] | CMOS-HV | V3P30 | Default | LAN0_PORT1_LED[1] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[40] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_LED[1] | O-OD | - | D |
| | | | | Func 2 | LAN0_PORT0_LED[3] | O-OD | - | D |
| R46 | LAN0_PORT1_RX_DN | KR | V1P05 | not shared | LAN0_PORT1_RX_DN | I, Diff | - | D |
| U46 | LAN0_PORT1_RX_DP | KR | V1P05 | not shared | LAN0_PORT1_RX_DP | I, Diff | - | D |
| AY66 | LAN0_PORT1_SDP[0] | CMOS-HV | V3P30 | Default | GPIO[15] | O | - | D |
| | | | | Func 0 GPIO | GPIO[15] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_SDP[0] | I,O | - | D |
| BB72 | LAN0_PORT1_SDP[1] | CMOS-HV | V3P30 | Default | GPIO[17] | O | - | D |
| | | | | Func 0 GPIO | GPIO[17] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_SDP[1] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT0_I2C_DATA | I,O-OD | - | D |
| AU65 | LAN0_PORT1_SDP[2] | CMOS-HV | V3P30 | Default | GPIO[19] | O | - | D |
| | | | | Func 0 GPIO | GPIO[19] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_SDP[2] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT1_SDP[0] | I,O | - | D |
| W58 | LAN0_PORT1_SDP[3] | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[21] | O | - | D |
| | | | | Func 0 GPIO | GPIO[21] | SW | SW | D |
| | | | | Func 1 | LAN0_PORT1_SDP[3] | I,O | - | D |
| | | | | Func 2 | LAN1_PORT1_I2C_DATA | I,O-OD | - | D |
| | | | | Func 3 | LAN1_PORT1_SDP[1] | I,O | - | D |
| G44 | LAN0_PORT1_TX_DN | KR | V1P05 | not shared | LAN0_PORT1_TX_DN | O, Diff | - | D |
| J44 | LAN0_PORT1_TX_DP | KR | V1P05 | not shared | LAN0_PORT1_TX_DP | O, Diff | - | D |
| AA50 | LAN0_RBIAS | KR | V1P05 | not shared | LAN0_RBIAS | I,O | - | D |
| AY70 | LAN1_PORT0_LED[0] | CMOS-HV | V3P30 | Default | LAN1_PORT0_LED[0] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[22] | SW | SW | D |
| | | | | Func 1 | LAN1_PORT0_LED[0] | O-OD | - | D |
| AY72 | LAN1_PORT0_LED[1] | CMOS-HV | V3P30 | Default | LAN1_PORT0_LED[1] | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[23] | SW | SW | D |
| | | | | Func 1 | LAN1_PORT0_LED[1] | O-OD | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 23 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| T37 | LAN1_PORT0_RX_DN | KR | V1P05 | not shared | LAN1_PORT0_RX_DN | I, Diff | - | D |
| P37 | LAN1_PORT0_RX_DP | KR | V1P05 | not shared | LAN1_PORT0_RX_DP | I, Diff | - | D |
| G33 | LAN1_PORT0_TX_DN | KR | V1P05 | not shared | LAN1_PORT0_TX_DN | O, Diff | - | D |
| J33 | LAN1_PORT0_TX_DP | KR | V1P05 | not shared | LAN1_PORT0_TX_DP | O, Diff | - | D |
| P40 | LAN1_PORT1_RX_DN | KR | V1P05 | not shared | LAN1_PORT1_RX_DN | I, Diff | - | D |
| T40 | LAN1_PORT1_RX_DP | KR | V1P05 | not shared | LAN1_PORT1_RX_DP | I, Diff | - | D |
| G37 | LAN1_PORT1_TX_DN | KR | V1P05 | not shared | LAN1_PORT1_TX_DN | O, Diff | - | D |
| J37 | LAN1_PORT1_TX_DP | KR | V1P05 | not shared | LAN1_PORT1_TX_DP | O, Diff | - | D |
| Y39 | LAN1_RBIAS | KR | V1P05 | not shared | LAN1_RBIAS | I,O | - | D |
| BN52 | MCERR_N | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PD | D |
| | | | | Default | MCERR_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[109] | SW | SW | D |
| | | | | Func 1 | MCERR_N | O | - | D |
| BL70 | MEMHOT_N | CMOS-LV | V1P05 | Default | MEMHOT_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[52] | SW | SW | D |
| | | | | Func 1 | MEMHOT_N | I | 20K PU | D |
| AU58 | NCSI_ARB_IN | CMOS-HV | V3P30 | Default | GPIO[32] | I | 20K PD | D |
| | | | | Func 0 GPIO | GPIO[32] | SW | SW | D |
| | | | | Func 1 | Reserved | I | 20K PD | |
| | | | | Func 2 | NCSI_ARB_IN | I | 20K PD | D |
| AV63 | NCSI_ARB_OUT | CMOS-HV | V3P30 | Default | GPIO[36] | I | - | D |
| | | | | Func 0 GPIO | GPIO[36] | SW | SW | D |
| | | | | Func 2 | NCSI_ARB_OUT | O | - | D |
| AY63 | NCSI_CLK_IN | CMOS-HV | V3P30 | Default | GPIO[29] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[29] | SW | SW | D |
| | | | | Func 2 | NCSI_CLK_IN | I | 20K PU | D |
| BD65 | NCSI_CRS_DV | CMOS-HV | V3P30 | Default | GPIO[31] | I | - | D |
| | | | | Func 0 GPIO | GPIO[31] | SW | SW | D |
| | | | | Func 2 | NCSI_CRS_DV | O-Tri | - | D |
| AY59 | NCSI_RXD[0] | CMOS-HV | V3P30 | Default | GPIO[28] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[28] | SW | SW | D |
| | | | | Func 2 | NCSI_RXD[0] | O-Tri | 20K PU | D |
| BD72 | NCSI_RXD[1] | CMOS-HV | V3P30 | Default | GPIO[30] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[30] | SW | SW | D |
| | | | | Func 2 | NCSI_RXD[1] | O-Tri | 20K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 24 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BD70 | NCSI_TX_EN | CMOS-HV | V3P30 | Default | GPIO[33] | I | - | D |
| | | | | Func 0 GPIO | GPIO[33] | SW | SW | D |
| | | | | Func 2 | NCSI_TX_EN | I | - | D |
| BE66 | NCSI_TXD[0] | CMOS-HV | V3P30 | Default | GPIO[34] | I | - | D |
| | | | | Func 0 GPIO | GPIO[34] | SW | SW | D |
| | | | | Func 2 | NCSI_TXD[0] | I | - | D |
| AL56 | NCSI_TXD[1] | CMOS-HV | V3P30 | Default | GPIO[35] | I | - | D |
| | | | | Func 0 GPIO | GPIO[35] | SW | SW | D |
| | | | | Func 2 | NCSI_TXD[1] | I | - | D |
| AP56 | PCIE_CLKREQ_N[0] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[0] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[41] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[0] | I,O-OD | - | D |
| BA58 | PCIE_CLKREQ_N[1] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[1] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[42] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[1] | I,O-OD | - | D |
| BH71 | PCIE_CLKREQ_N[2] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[2] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[43] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[2] | I,O-OD | - | D |
| BJ66 | PCIE_CLKREQ_N[3] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[3] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[44] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[3] | I,O-OD | - | D |
| BD61 | PCIE_CLKREQ_N[4] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[4] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[45] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[4] | I,O-OD | - | D |
| | | | | Func 2 | FAN_TACH[3] | I | - | D |
| BW52 | PCIE_CLKREQ_N[5] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[5] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[98] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[5] | I,O-OD | - | D |
| | | | | Func 2 | UART2_RTS | O | - | D |
| | | | | Func 3 | UART0_RTS | O | - | D |
| BV51 | PCIE_CLKREQ_N[6] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[6] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[99] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[6] | I,O-OD | - | D |
| | | | | Func 2 | UART2_RXD | I | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 25 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BK58 | PCIE_CLKREQ_N[7] | CMOS-HV | V3P30 | Default | PCIE_CLKREQ_N[7] | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[100] | SW | SW | D |
| | | | | Func 1 | PCIE_CLKREQ_N[7] | I,O-OD | - | D |
| | | | | Func 2 | UART2_TXD | O | - | D |
| | | | | Func 3 | LAN1_PORT1_LED[0] | O-OD | - | D |
| BL63 | PMU_PLTRST_N | CMOS-HV | V3P30 | Default | PMU_PLTRST_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[87] | SW | SW | D |
| | | | | Func 1 | PMU_PLTRST_N | O | - | D |
| CA59 | PMU_PWRBTN_N | CMOS-HV | V3P30 | Default | PMU_PWRBTN_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[85] | SW | SW | D |
| | | | | Func 1 | PMU_PWRBTN_N | I | 20K PU | D |
| BL59 | PMU_RESETBUTTON_N | CMOS-HV | V3P30 | Default | PMU_RESETBUTTON_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[86] | SW | SW | D |
| | | | | Func 1 | PMU_RESETBUTTON_N | I | 20K PU | D |
| BK65 | PMU_SLP_S3_N | CMOS-HV | V3P30 | Default | PMU_SLP_S3_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[83] | SW | SW | D |
| | | | | Func 1 | PMU_SLP_S3_N | O | - | D |
| BW59 | PMU_SLP_S45_N | CMOS-HV | V3P30 | Default | PMU_SLP_S45_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[82] | SW | SW | D |
| | | | | Func 1 | PMU_SLP_S45_N | O | - | D |
| BG59 | PMU_SUSCLK | CMOS-HV | V3P30 | Default | PMU_SUSCLK | O | - | D |
| | | | | Func 0 GPIO | GPIO[80] | SW | SW | D |
| | | | | Func 1 | PMU_SUSCLK | O | - | D |
| BR61 | PMU_WAKE_N | CMOS-HV | V3P30 | Default | PMU_WAKE_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[84] | SW | SW | D |
| | | | | Func 1 | PMU_WAKE_N | I | 20K PU | D |
| BM71 | PROCHOT_N | CMOS-LV | V1P05 | Default | PROCHOT_N | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[51] | SW | SW | D |
| | | | | Func 1 | PROCHOT_N | I,O-OD | 20K PU | D |
| AE59 | RSMRST_N | RTC | VRTC | not shared | RSMRST_N | I | - | D |
| H35 | RSVD_H35 | n/a | - | not shared | RSVD_H35 | O | - | D |
| H42 | RSVD_H42 | n/a | - | not shared | RSVD_H42 | O | - | D |
| K35 | RSVD_K35 | n/a | - | not shared | RSVD_K35 | O | - | D |
| K42 | RSVD_K42 | n/a | - | not shared | RSVD_K42 | O | - | D |
| P33 | RSVD_P33 | n/a | - | not shared | RSVD_P33 | I,O | - | D |
| R31 | RSVD_R31 | n/a | - | not shared | RSVD_R31 | O | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 26 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| R50 | RSVD_R50 | n/a | - | not shared | RSVD_R50 | I | - | D |
| R54 | RSVD_R54 | n/a | - | not shared | RSVD_R54 | O, Diff | - | D |
| R58 | RSVD_R58 | n/a | - | not shared | RSVD_R58 | O, Diff | - | D |
| T33 | RSVD_T33 | n/a | - | not shared | RSVD_T33 | I,O | - | D |
| T56 | RSVD_T56 | n/a | - | not shared | RSVD_T56 | O, Diff | - | D |
| T59 | RSVD_T59 | n/a | - | not shared | RSVD_T59 | O, Diff | - | D |
| U31 | RSVD_U31 | n/a | - | not shared | RSVD_U31 | O | - | D |
| W50 | RSVD_W50 | n/a | - | not shared | RSVD_W50 | I,O | - | D |
| Y48 | RSVD_Y48 | n/a | - | not shared | RSVD_Y48 | I,O | - | D |
| AA29 | RSVD_AA29 | n/a | - | not shared | RSVD_AA29 | O | - | D |
| AD58 | RSVD_AD58 | n/a | - | not shared | RSVD_AD58 | I,O | - | D |
| AH20 | RSVD_AH20 | n/a | - | not shared | RSVD_AH20 | O, Diff | - | D |
| AH54 | RSVD_AH54 | n/a | - | not shared | RSVD_AH54 | I | - | D |
| AJ22 | RSVD_AJ22 | n/a | - | not shared | RSVD_AJ22 | O, Diff | - | D |
| AK20 | RSVD_AK20 | n/a | - | not shared | RSVD_AK20 | O, Diff | - | D |
| AL22 | RSVD_AL22 | n/a | - | not shared | RSVD_AL22 | O, Diff | - | D |
| AL59 | RSVD_AL59 | n/a | - | not shared | RSVD_AL59 | I | - | D |
| AN20 | RSVD_AN20 | n/a | - | not shared | RSVD_AN20 | O | - | D |
| AP22 | RSVD_AP22 | n/a | - | not shared | RSVD_AP22 | O | - | D |
| AW30 | RSVD_AW30 | n/a | - | not shared | RSVD_AW30 | O | - | D |
| BA24 | RSVD_BA24 | n/a | - | not shared | RSVD_BA24 | I,O | - | D |
| BA26 | RSVD_BA26 | n/a | - | not shared | RSVD_BA26 | I,O | - | D |
| BA29 | RSVD_BA29 | n/a | - | not shared | RSVD_BA29 | O | - | D |
| BB29 | RSVD_BB29 | n/a | - | not shared | RSVD_BB29 | O | - | D |
| BF38 | RSVD_BF38 | n/a | - | not shared | RSVD_BF38 | O | - | D |
| BG40 | RSVD_BG40 | n/a | - | not shared | RSVD_BG40 | I | - | D |
| BH38 | RSVD_BH38 | n/a | - | not shared | RSVD_BH38 | I,O | - | D |
| BK38 | RSVD_BK38 | n/a | - | not shared | RSVD_BK38 | I,O | - | D |
| BY58 | RSVD_BY58 | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | RSVD_BY58 | O | - | D |
| | | | | Func 0 GPIO | GPIO[89] | SW | SW | D |
| | | | | Func 1 | RSVD_BY58 | O | - | |
| AG59 | RTC_X1_PAD | Analog | V1P80 | not shared | RTC_X1_PAD | I,O | - | D |
| AH58 | RTC_X2_PAD | Analog | V1P80 | not shared | RTC_X2_PAD | I,O | - | D |
| AJ56 | RTEST_N | RTC | VRTC | not shared | RTEST_N | I | - | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 27 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| CA57 | SATA_PDETECT[0] | CMOS-HV | V3P30 | Default | SATA_PDETECT[0] | I | - | D |
| | | | | Func 0 GPIO | GPIO[92] | SW | SW | D |
| | | | | Func 1 | SATA_PDETECT[0] | I | - | D |
| | | | | Func 2 | SATA0_SLD | O-OD | - | D |
| BN63 | SATA_PDETECT[1] | CMOS-HV | V3P30 | Default | SATA_PDETECT[1] | I | - | D |
| | | | | Func 0 GPIO | GPIO[93] | SW | SW | D |
| | | | | Func 1 | SATA_PDETECT[1] | I | - | D |
| | | | | Func 2 | SATA1_SLD | O-OD | - | D |
| BR65 | SATA0_LED_N | CMOS-HV | V3P30 | Default | SATA0_LED_N | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[90] | SW | SW | D |
| | | | | Func 1 | SATA0_LED_N | O-OD | - | D |
| | | | | Func 2 | SATA0_SCLK | O-OD | - | D |
| BU57 | SATA0_SDOUT | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[94] | O | - | D |
| | | | | Func 0 GPIO | GPIO[94] | SW | SW | D |
| | | | | Func 1 | UART1_RTS | O | - | D |
| | | | | Func 2 | SATA0_SDOUT | O-OD | - | D |
| | | | | Func 3 | UART_IE_RTS | O | - | D |
| BV55 | SATA1_LED_N | CMOS-HV | V3P30 | Default | SATA1_LED_N | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[91] | SW | SW | D |
| | | | | Func 1 | SATA1_LED_N | O-OD | - | D |
| | | | | Func 2 | SATA1_SCLK | O-OD | - | D |
| BW54 | SATA1_SDOUT | CMOS-HV | V3P30 | Default | GPIO[95] | O | - | D |
| | | | | Func 0 GPIO | GPIO[95] | SW | SW | D |
| | | | | Func 1 | UART1_CTS | I | - | D |
| | | | | Func 2 | SATA1_SDOUT | O-OD | - | D |
| | | | | Func 3 | UART_IE_CTS | I | - | D |
| BR50 | SMB_HOST_CLK | CMOS-HV | V3P30 | Default | SMB_HOST_CLK | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[114] | SW | SW | D |
| | | | | Func 1 | SMB_HOST_CLK | I,O-OD | 20K PU | D |
| | | | | Func 2 | SMB_ME_SMT2_CLK | I,O-OD | 20K PU | D |
| | | | | Func 3 | SMB_IE_SMT1_CLK | I,O-OD | 20K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 28 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BK50 | SMB_HOST_DATA | CMOS-HV | V3P30 | Default | SMB_HOST_DATA | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[113] | SW | SW | D |
| | | | | Func 1 | SMB_HOST_DATA | I,O-OD | 20K PU | D |
| | | | | Func 2 | SMB_ME_SMT2_DATA | I,O-OD | 20K PU | D |
| | | | | Func 3 | SMB_IE_SMT1_DATA | I,O-OD | 20K PU | D |
| BN66 | SMB_LAN_ALRT_N | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[13] | O | - | D |
| | | | | Func 0 GPIO | GPIO[13] | SW | SW | D |
| | | | | Func 1 | UART2_CTS | I | - | D |
| | | | | Func 2 | UART0_CTS | I | - | D |
| | | | | Func 3 | SMB_LAN_ALRT_N | I,O-OD | - | D |
| BH54 | SMB_LAN_CLK | CMOS-HV | V3P30 | Default | GPIO[103] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[103] | SW | SW | D |
| | | | | Func 1 | CPU_RESET_N | O | 20K PU | D |
| | | | | Func 2 | FAN_PWM[3] | O-OD | 20K PU | D |
| | | | | Func 3 | SMB_LAN_CLK | I,O-OD | 20K PU | D |
| BK54 | SMB_LAN_DATA | CMOS-HV | V3P30 | Default | GPIO[104] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[104] | SW | SW | D |
| | | | | Func 3 | SMB_LAN_DATA | I,O-OD | 20K PU | D |
| BW48 | SMB_LEG_ALRT_N | CMOS-HV | V3P30 | Default | SMB_LEG_ALRT_N | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[112] | SW | SW | D |
| | | | | Func 1 | SMB_LEG_ALRT_N | I,O-OD | 20K PU | D |
| BY51 | SMB_LEG_CLK | CMOS-HV | V3P30 | Default | SMB_LEG_CLK | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[110] | SW | SW | D |
| | | | | Func 1 | SMB_LEG_CLK | I,O-OD | 20K PU | D |
| BV49 | SMB_LEG_DATA | CMOS-HV | V3P30 | Default | SMB_LEG_DATA | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[111] | SW | SW | D |
| | | | | Func 1 | SMB_LEG_DATA | I,O-OD | 20K PU | D |
| BG52 | SMB_ME_SMT0_ALRT_N | CMOS-HV | V3P30 | Default | SMB_ME_SMT0_ALRT_N | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[119] | SW | SW | D |
| | | | | Func 1 | SMB_ME_SMT0_ALRT_N | I,O-OD | 20K PU | D |
| BR46 | SMB_ME_SMT0_CLK | CMOS-HV | V3P30 | Default | SMB_ME_SMT0_CLK | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[118] | SW | SW | D |
| | | | | Func 1 | SMB_ME_SMT0_CLK | I,O-OD | 20K PU | D |
| | | | | Func 2 | SMB_IE_SMT2_CLK | I,O-OD | 20K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 29 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BV47 | SMB_ME_SMT0_DATA | CMOS-HV | V3P30 | Default | SMB_ME_SMT0_DATA | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[117] | SW | SW | D |
| | | | | Func 1 | SMB_ME_SMT0_DATA | I,O-OD | 20K PU | D |
| | | | | Func 2 | SMB_IE_SMT2_DATA | I,O-OD | 20K PU | D |
| BR54 | SMB_PECI_CLK | CMOS-HV | V3P30 | Default | SMB_PECI_CLK | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[116] | SW | SW | D |
| | | | | Func 1 | SMB_PECI_CLK | I,O-OD | 20K PU | D |
| BY49 | SMB_PECI_DATA | CMOS-HV | V3P30 | Default | SMB_PECI_DATA | I,O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[115] | SW | SW | D |
| | | | | Func 1 | SMB_PECI_DATA | I,O-OD | 20K PU | D |
| CA61 | SPI_CLK | CMOS-HV | V3P30 | Default | SPI_CLK | I,O | 20K PU or 20K PD | D |
| | | | | Func 0 GPIO | GPIO[78] | SW | SW | D |
| | | | | Func 1 | SPI_CLK | I,O | 20K PU or 20K PD | D |
| BG56 | SPI_CS_N[0] | CMOS-HV | V3P30 | Default | SPI_CS_N[0] | I,O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[72] | SW | SW | D |
| | | | | Func 1 | SPI_CS_N[0] | I,O | 20K PU | D |
| BL66 | SPI_CS_N[1] | CMOS-HV | V3P30 | Default | SPI_CS_N[1] | I,O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[73] | SW | SW | D |
| | | | | Func 1 | SPI_CS_N[1] | I,O | 20K PU | D |
| BH61 | SPI_IO[2] | CMOS-HV | V3P30 | Default | SPI_IO[2] | I,O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[76] | SW | SW | D |
| | | | | Func 1 | SPI_IO[2] | I,O | 20K PU | D |
| BW61 | SPI_IO[3] | CMOS-HV | V3P30 | Default | SPI_IO[3] | I,O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[77] | SW | SW | D |
| | | | | Func 1 | SPI_IO[3] | I,O | 20K PU | D |
| BV62 | SPI_MISO_IO[1] | CMOS-HV | V3P30 | Default | SPI_MISO_IO[1] | I,O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[75] | SW | SW | D |
| | | | | Func 1 | SPI_MISO_IO[1] | I,O | 20K PU | D |
| BU63 | SPI_MOSI_IO[0] | CMOS-HV | V3P30 | Default | SPI_MOSI_IO[0] | I,O | 20K PU or 20K PD | D |
| | | | | Func 0 GPIO | GPIO[74] | SW | SW | D |
| | | | | Func 1 | SPI_MOSI_IO[0] | I,O | 20K PU or 20K PD | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 30 of 31)

| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| Y59 | SRTCRST_N | RTC | VRTC | not shared | SRTCRST_N | I | - | D |
| BV58 | SUS_STAT_N | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | SUS_STAT_N | O | - | D |
| | | | | Func 0 GPIO | GPIO[88] | SW | SW | D |
| | | | | Func 1 | SUS_STAT_N | O | - | D |
| BH58 | SUSPWRDNACK | CMOS-HV | V3P30 | Default | SUSPWRDNACK | O | - | D |
| | | | | Func 0 GPIO | GPIO[79] | SW | SW | D |
| | | | | Func 1 | SUSPWRDNACK | O | - | D |
| BA54 | SVID_ALERT_N | CMOS-LV | V1P05 | Default | SVID_ALERT_N | I | - | D |
| | | | | Func 0 GPIO | GPIO[47] | SW | SW | D |
| | | | | Func 1 | SVID_ALERT_N | I | - | D |
| BE59 | SVID_CLK | CMOS-LV | V1P05 | Default | SVID_CLK | O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[49] | SW | SW | D |
| | | | | Func 1 | SVID_CLK | O-OD | - | D |
| BE56 | SVID_DATA | CMOS-LV | V1P05 | Default | SVID_DATA | I,O-OD | - | D |
| | | | | Func 0 GPIO | GPIO[48] | SW | SW | D |
| | | | | Func 1 | SVID_DATA | I,O-OD | - | D |
| BN40 | TCK | CMOS-LV | V1P05 | Default | TCK | I | 20K PD | D |
| | | | | Func 0 GPIO | Cannot be a Customer GPIO | - | - | - |
| | | | | Func 1 | TCK | I | 20K PD | D |
| BG44 | TDI | CMOS-LV | V1P05 | Default | TDI | I | 20K PU | D |
| | | | | Func 0 GPIO | Cannot be a Customer GPIO | - | - | - |
| | | | | Func 1 | TDI | I | 20K PU | D |
| BV38 | TDO | CMOS-LV | V1P05 | Default | TDO | I,O-OD | - | D |
| | | | | Func 0 GPIO | Cannot be a Customer GPIO | - | - | - |
| | | | | Func 1 | TDO | I,O-OD | - | D |
| BA61 | THERMTRIP_N | CMOS-LV | V1P05 | Default | THERMTRIP_N | O-OD | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[50] | SW | SW | D |
| | | | | Func 1 | THERMTRIP_N | O-OD | 20K PU | D |
| BW39 | TMS | CMOS-LV | V1P05 | Default | TMS | I | 20K PU | D |
| | | | | Func 0 GPIO | Cannot be a Customer GPIO | - | - | - |
| | | | | Func 1 | TMS | I | 20K PU | D |
| BL40 | TRST_N | CMOS-LV | V1P05 | Default | TRST_N | I | 20K PU | D |
| | | | | Func 0 GPIO | Cannot be a Customer GPIO | - | - | - |
| | | | | Func 1 | TRST_N | I | 20K PU | D |

## Table 39-1. Signal Pin Name Directory - Sorted by Name (Sheet 31 of 31)

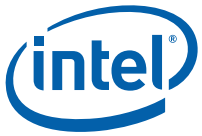
| Pin | Pin Name | Buffer Type | V Group | Pin States | Signal Name | Dir | Internal PU/PD | Description (Link) |
|---|---|---|---|---|---|---|---|---|
| BL56 | UART0_RXD | CMOS-HV | V3P30 | Default | GPIO[101] | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[101] | SW | SW | D |
| | | | | Func 1 | UART0_RXD | I | 20K PU | D |
| CA52 | UART0_TXD | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PD | D |
| | | | | Default | GPIO[102] | O | - | D |
| | | | | Func 0 GPIO | GPIO[102] | SW | SW | D |
| | | | | Func 1 | UART0_TXD | O | - | D |
| BV53 | UART1_RXD | CMOS-HV | V3P30 | Default | GPIO[96] | O | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[96] | SW | SW | D |
| | | | | Func 1 | UART1_RXD | I | 20K PU | D |
| | | | | Func 2 | UART_IE_RXD | I | 20K PU | D |
| BN59 | UART1_TXD | CMOS-HV | V3P30 | Strap Sampling | Reserved for Intel | I | 20K PU | D |
| | | | | Default | GPIO[97] | O | - | D |
| | | | | Func 0 GPIO | GPIO[97] | SW | SW | D |
| | | | | Func 1 | UART1_TXD | O | - | D |
| | | | | Func 2 | UART_IE_TXD | O | - | D |
| BD54 | USB_OC_N | CMOS-HV | V3P30 | Default | USB_OC_N | I | 20K PU | D |
| | | | | Func 0 GPIO | GPIO[120] | SW | SW | D |
| | | | | Func 1 | USB_OC_N | I | 20K PU | D |
| AN54 | USB2_COMP | USB2 | V3P30 | not shared | USB2_COMP | I | - | D |
| AR72 | USB2_DN[0] | USB2 | V3P30 | not shared | USB2_DN[0] | I,O, Diff | - | D |
| AP63 | USB2_DN[1] | USB2 | V3P30 | not shared | USB2_DN[1] | I,O, Diff | - | D |
| AV72 | USB2_DN[2] | USB2 | V3P30 | not shared | USB2_DN[2] | I,O, Diff | - | D |
| AN61 | USB2_DN[3] | USB2 | V3P30 | not shared | USB2_DN[3] | I,O, Diff | - | D |
| AR70 | USB2_DP[0] | USB2 | V3P30 | not shared | USB2_DP[0] | I,O, Diff | - | D |
| AR65 | USB2_DP[1] | USB2 | V3P30 | not shared | USB2_DP[1] | I,O, Diff | - | D |
| AV70 | USB2_DP[2] | USB2 | V3P30 | not shared | USB2_DP[2] | I,O, Diff | - | D |
| AP59 | USB2_DP[3] | USB2 | V3P30 | not shared | USB2_DP[3] | I,O, Diff | - | D |

### 39.1.1 Reserved (RSVD_xxx) Pins

A number of the SoC pins are reserved for Intel. Table 39-2 lists these pins and provides recommendations for the platform board design.

**Table 39-2. Treatment of Reserved Pins (Sheet 1 of 2)**

| Pin | Pin Name | Dir | Shared-Function Pin? | Voltage Rail | SoC Has Internal Resistor | Board Design Recommendation |
|---|---|---|---|---|---|---|
| H35 | RSVD_H35 | O | No | n/a | n/a | Make no connection to this pin. |
| H42 | RSVD_H42 | O | No | n/a | n/a | Make no connection to this pin. |
| K35 | RSVD_K35 | O | No | n/a | n/a | Make no connection to this pin. |
| K42 | RSVD_K42 | O | No | n/a | n/a | Make no connection to this pin. |
| P33 | RSVD_P33 | I,O | No | n/a | n/a | Make no connection to this pin. |
| R31 | RSVD_R31 | O | No | n/a | n/a | Make no connection to this pin. |
| R50 | RSVD_R50 | I | No | VSS | No | Use Pull-Down Resistor with a value ranging from 0 to 1 kΩ, ±5% |
| R54 | RSVD_R54 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| R58 | RSVD_R58 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| T33 | RSVD_T33 | I,O | No | n/a | n/a | Make no connection to this pin. |
| T56 | RSVD_T56 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| T59 | RSVD_T59 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| U31 | RSVD_U31 | O | No | n/a | n/a | Make no connection to this pin. |
| W50 | RSVD_W50 | I,O | No | n/a | n/a | Make no connection to this pin. |
| Y48 | RSVD_Y48 | I,O | No | n/a | n/a | Make no connection to this pin. |
| AA29 | RSVD_AA29 | O | No | n/a | n/a | Make no connection to this pin. |
| AD58 | RSVD_AD58 | I,O | No | n/a | n/a | Make no connection to this pin. |
| AH20 | RSVD_AH20 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| AH54 | RSVD_AH54 | I | No | VSS | No | Use Pull-Down Resistor with a value ranging from 0 to 1 kΩ, ±5% |
| AJ22 | RSVD_AJ22 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| AK20 | RSVD_AK20 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| AL22 | RSVD_AL22 | O, Diff | No | n/a | n/a | Make no connection to this pin. |
| AL59 | RSVD_AL59 | I | No | VSS | No | Use Pull-Down Resistor with a value ranging from 0 to 1 kΩ, ±5% |
| AN20 | RSVD_AN20 | O | No | n/a | n/a | Make no connection to this pin. |
| AP22 | RSVD_AP22 | O | No | n/a | n/a | Make no connection to this pin. |
| AW30 | RSVD_AW30 | O | No | n/a | n/a | Make no connection to this pin. |
| BA24 | RSVD_BA24 | I,O | No | n/a | n/a | Make no connection to this pin. |
| BA26 | RSVD_BA26 | I,O | No | n/a | n/a | Make no connection to this pin. |
| BA29 | RSVD_BA29 | O | No | n/a | n/a | Make no connection to this pin. |
| BB29 | RSVD_BB29 | O | No | n/a | n/a | Make no connection to this pin. |
| BF38 | RSVD_BF38 | O | No | n/a | n/a | Make no connection to this pin. |
| BG40 | RSVD_BG40 | I | No | VSS | No | Use Pull-Down Resistor with a value ranging from 0 to 1 kΩ, ±5% |



## Table 39-2. Treatment of Reserved Pins (Sheet 2 of 2)

| Pin | Pin Name | Dir | Shared-Function Pin? | Voltage Rail | SoC Has Internal Resistor | Board Design Recommendation |
|---|---|---|---|---|---|---|
| BH38 | RSVD_BH38 | I,O | No | n/a | n/a | Make no connection to this pin. |
| BK38 | RSVD_BK38 | I,O | No | n/a | n/a | Make no connection to this pin. |
| BY58 | RSVD_BY58 | O | Yes | V3P30 | No | Can be programmed to be used as Customer GPIO[89] signal pin. If not used as a GPIO, make no connection to this pin. |

## 39.2 Directory of Signals that Share the HSIO Pins

The High-Speed I/O (HSIO) signals listed in Table 39-1, "Signal Pin Name Directory - Sorted by Name" are shared with PCI Express*, SATA, and USB 3.0 signal pins.

Table 39-1, "Signal Pin Name Directory - Sorted by Name" lists these shared signal names.

The individual shared signals are described in the following chapters:

- Chapter 11, "PCI Express Root Ports (RP)"
- Chapter 12, "SATA Controller"
- Chapter 14, "USB Combo Controller"

Readers of the electronic version of this document can navigate to these description chapters by clicking on the blue-colored "D" in the Description column of Table 39-3, "Signals that Share the Twenty HSIO Lanes."

The association of the Signals in Table 39-3, "Signals that Share the Twenty HSIO Lanes" with the 20 HSIO Lanes, the supported shared patterns, and the HSIO configuration control, are described in Chapter 10, "Flexible I/O Adapter (FIA) Overview."

**Table 39-3. Signals that Share the Twenty HSIO Lanes**

| Signal Name | Direction | Link to Signal Description |
|---|---|---|
| **PCI Express Root Port Cluster 0 (RP0, RP1, RP2, RP3) - 8 Lanes** | | |
| PCIE0_RP_3_0_TX_DP[7:0]<br>PCIE0_RP_3_0_TX_DN[7:0] | O, Differential | D |
| PCIE0_RP_3_0_RX_DP[7:0]<br>PCIE0_RP_3_0_RX_DN[7:0] | I, Differential | D |
| **PCI Express Root Port Cluster 1 (RP4, RP5, RP6, RP7) - 8 Lanes** | | |
| PCIE1_RP_7_4_TX_DP[7:0]<br>PCIE1_RP_7_4_TX_DN[7:0] | O, Differential | D |
| PCIE1_RP_7_4_RX_DP[7:0]<br>PCIE1_RP_7_4_RX_DN[7:0] | I, Differential | D |
| **SATA0 Controller - 8 Lanes** | | |
| SATA0_TX_DP[7:0]<br>SATA0_TX_DN[7:0] | O, Differential | D |
| SATA0_RX_DP[7:0]<br>SATA0_RX_DN[7:0] | I, Differential | D |
| **SATA1 Controller - 8 Lanes** | | |
| SATA1_TX_DP[7:0]<br>SATA1_TX_DN[7:0] | O, Differential | D |
| SATA1_RX_DP[7:0]<br>SATA1_RX_DN[7:0] | I, Differential | D |
| **USB 3.0 Interface of the USB Combo Controller - 4 Lanes** | | |
| USB3_TX_DP[3:0]<br>USB3_TX_DN[3:0] | O, Differential | D |
| USB3_RX_DP[3:0]<br>USB3_RX_DN[3:0] | I, Differential | D |

## 39.3 Directory of Voltage Supply and Sense Pin Names

Special on-board filtering is required.

**Table 39-4. Voltage Pins Sorted by Platform VR and SoC Voltage Group (Sheet 1 of 3)**

| Platform VR | SoC Voltage Group | Pin Number(s) | Power Delivery Notes |
|---|---|---|---|
| Voltage Observe Output Pins | OBS_VCCRAMCPUSI1GT_MOD0 | AA37 | |
| | OBS_VCCCPUVIDSI0GT_MOD0 | Y36 | |
| | OBS_VCCCPUVIDSI0GT_MOD4 | BA36 | |
| | OBS_VCCBIUVIDSI0GT_MOD0 | AA36 | |
| Voltage Sense Output Pins | SENSE_VCCRAMCPU_S0 | AW36 | |
| | SENSE_VSSRAMCPU_S0 | AW37 | |
| | SENSE_VCCCPUVIDSI0 | AE37 | |
| | SENSE_VSSCPUVIDSI0 | AE39 | |
| V1P05 | VCCACKDDR_1P05 | AA27, AC27, AE27, AG27, AJ27, AK27, AM27, AP27, AT27, AU27, AW27 | |
| | VCCADDR_1P05 | AA26, AC26, AE26, AG26, AJ26, AK26, AM26, AP26, AT26, AU26, AW26 | |
| | VCCAON_HSIO_1P05 | AF50, AG52, AH50, AK50, AN50, AP48, AR50, AT48 | |
| | VCCCL_HSIO_1P05 | AJ48, AK48 | |
| | VCCFHVSI0_GROUP0_1P05 | BD38 | |
| | VCCFHVSI0_GROUP1_1P05 | AW39 | |
| | VCCFHVSI0_GROUP2_1P05 | AW41 | |
| | VCCFHVSI0_1P05 | AU39 | |
| | VCCPLLEBB_HSIO_1P05 | AG48 | |
| | VCCKRLCPLL | AA41, AA43, AA45, AA46 | 1 |
| | VCCKRA_LV | AC39, AC41, AC43, AC45, AC46, AC48, AE41, AE43, AE45, AE46 | |
| | VCCUSB_1P05 | AR54, AU54 | |
| | VCCDIGICKSI0_1P05 | AG45, AJ45 | 1 |
| | VCCDIGXXXSI0_1P05 | AK45, AM45, AP45, AT45 | |
| V1P80 | VCCKR_HV | Y43, Y45 | 1 |
| | VCCUSBSUS_1P8 | AU50, AW50 | |
| | VCCPADXXXSI0_1P8 | AW43, BA43 | |
| | VCCFHVIFPSI0_1P8 | BA39 | |
| V3P30 (LPC) | VCC_LPC_ESPI_3P3_1P8 | BD46 | |
| V3P30 | VCCUSBSUS_3P3 | AW48, BA48 | |
| | VCCPADXXXSI0_3P3 | AW46, BA46 | |
| | VCCFHVIFPSI0_3P3 | BA41 | |
| VRTC | VCCRTC_3P3_G3 | AU48 | |
| VCC | VCCCPU_SVID_S0 | AA32, AA34, AC32, AC34, AE32, AE34, AG32, AG34, AJ32, AJ34, AK32, AK34, AM32, AM34, AP32, AP34, AT32, AT34, AU32, AU34, AW32, AW34, BA32, BA34, BB33, BD35, BE33, BF35, BG33, BH35, BJ33, BK35, BL33, BM35, BN33, BR35, BU32, BU34, BV33, BW32, BW34, BY33, CA32, CA34 | |

## Table 39-4. Voltage Pins Sorted by Platform VR and SoC Voltage Group (Sheet 2 of 3)

| Platform VR | SoC Voltage Group | Pin Number(s) | Power Delivery Notes |
|---|---|---|---|
| VCCRAM | VCCRAM_S0 | AC30, AC36, AE30, AE36, AG30, AG36, AJ30, AJ36, AK30, AK36, AM30, AM36, AP30, AP36, AT30, AT36, AU30, AU36 | |
| VCCREF | VCCUSB_1P24 | AT52, AV52 | |
| | VCCREF_DTS_E | AP29 | 1, 2 |
| | VCCREF_DTS_W | AU45 | 1 |
| | VCCREF_SFRXXXSI0_N | AA30 | 1 |
| | VCCREF_SFRXXXSI0_S | AW29 | 1 |
| VDDQ | VCCDDR_VDDQ | K9, M11, P11, P15, T11, V11, Y18, AC18, AE18, AG18, AJ18, AL18, AR1, AV1, AY1, BD1, BJ1, BJ3, BY12 | |
| | VCCCKDDR_VDDQ | AR24, AU24 | |
| | VCCDDRSFR_VDDQ | AH24, AK24 | 1 |
| | VCCDDR_VDDQ_NCTF | BN1, CA9 | |

## Table 39-4. Voltage Pins Sorted by Platform VR and SoC Voltage Group (Sheet 3 of 3)

| Platform VR | SoC Voltage Group | Pin Number(s) | Power Delivery Notes |
|---|---|---|---|
| VNN | VNN_SVID | AG39, AJ39, AJ41, AK39, AK41, AM39, AM41, AP39, AP41, AT39, AT41, AU41 | |
| VSS | VSS | A22, A37, B14, B29, B38, B40, B42, B44, B47, B49, B51, B53, B55, B58, B60, B62, B64, C5, C22, C37, C68, D14, D29, D38, D40, D42, D44, D47, D49, D51, D53, D55, D58, D60, D62, D64, D66, E3, E9, E11, E13, E15, E18, E20, E22, E24, E26, E28, E30, E32, E34, E37, E39, E41, E43, E45, E48, E50, E52, E54, E57, E59, E61, E63, E65, E67, E70, F3, F5, F8, F66, F70, G6, G29, G66, H4, H17, H31, H38, H46, H50, H58, J3, J29, J68, K2, K4, K17, K31, K38, K46, K54, K61, K69, K71, L5, L68, M18, M22, M26, M29, M33, M37, M40, M44, M66, M69, M71, N5, N17, N20, N24, N27, N31, N35, N38, N42, N46, N50, N54, N58, N61, N68, P29, P44, P48, P52, P56, P59, P63, P69, P71, R5, R17, R35, R38, R61, R68, T29, T44, T48, T52, T66, T69, T71, U17, U35, U38, U50, U54, U58, U61, V1, V3, V5, V18, V22, V26, V29, V33, V37, V40, V44, V48, V52, V56, V59, V63, V68, W9, W13, W20, W24, W61, W69, W71, Y5, Y7, Y11, Y22, Y26, Y27, Y29, Y30, Y32, Y34, Y37, Y41, Y46, Y52, Y56, Y66, Y68, AA13, AA24, AA39, AA48, AA54, AA58, AA61, AA69, AA71, AB5, AB68, AC29, AC37, AC52, AC63, AC69, AC71, AD5, AD13, AD50, AD61, AD68, AE2, AE4, AE29, AE48, AE52, AE56, AE66, AE69, AE71, AF5, AF13, AF24, AF54, AF58, AF61, AF68, AG29, AG37, AG41, AG43, AG46, AG63, AH13, AH61, AH69, AH71, AJ5, AJ29, AJ37, AJ43, AJ46, AJ52, AJ66, AJ68, AK9, AK13, AK29, AK37, AK43, AK46, AK54, AK58, AK61, AK69, AK71, AL5, AL7, AL11, AL15, AL52, AL63, AL68, AM29, AM37, AM43, AM46, AM48, AM69, AM71, AN1, AN3, AN5, AN13, AN17, AN24, AN58, AN65, AN68, AP18, AP37, AP43, AP46, AP52, AP66, AP69, AP71, AR5, AR13, AR58, AR61, AR68, AT18, AT29, AT37, AT43, AT46, AT56, AT59, AT63, AT66, AT69, AT71, AU29, AU37, AU43, AU46, AV5, AV18, AV68, AW24, AW45, AW54, AW58, AW61, AW65, AW69, AW71, AY5, AY18, AY68, BA9, BA13, BA20, BA27, BA30, BA37, BA45, BA50, BB5, BB7, BB11, BB18, BB22, BB26, BB37, BB40, BB44, BB48, BB52, BB56, BB59, BB63, BB68, BC71, BD5, BD9, BD13, BD20, BD24, BD27, BD31, BD68, BE7, BE15, BE18, BE37, BE71, BF5, BF13, BF17, BF31, BF42, BF46, BF50, BF54, BF58, BF61, BF65, BF68, BG18, BG37, BH31, BJ5, BJ11, BJ18, BJ22, BJ26, BJ29, BJ37, BJ40, BJ44, BJ48, BJ52, BJ56, BJ59, BJ63, BJ68, BJ70, BK20, BK24, BK27, BK31, BL5, BL18, BL37, BL68, BL72, BM31, BM42, BM46, BM50, BM54, BM58, BM61, BM65, BM69, BN5, BN11, BN18, BN37, BR3, BR6, BR31, BR38, BR70, BT8, BT66, BU3, BU9, BU11, BU13, BU15, BU18, BU20, BU22, BU24, BU26, BU28, BU30, BU37, BU39, BU41, BU45, BU48, BU50, BU52, BU54, BU59, BU61, BU65, BU67, BU70, BV16, BV31, BV35, BW24, BW43, BW57, BW63, BW65, BY16, BY31, BY35, BY40, BY47, BY53, BY60, CA24 | |
| | VSS_NCTF | A5, A7, A9, C3, C70, C72, E1, E72, G1, G72, J1, J72, A65, A66, A68, BN72, BR1, BR72, BU1, BU72, BW1, BW3, BW70, BW72, CA5, CA7, CA65, CA66, CA68, CA70 | |

## 39.4 Directory of Rotation-Detect Pins

### Table 39-5. Package Rotation (ROT) Pins

| Pin Name | Pin Number(s) | Description |
|---|---|---|
| NC_ROT | A70 | **Package Rotation (ROT) Detect**: For Intel use on special validation boards with a socket for the SoC package. NC_ROT has no connection inside the SoC package. VSS_ROT is connected to VSS power plane of the SoC package. For customer board designs, these pins can be left as no-connects or can be tied to VSS on the board. |
| VSS_ROT | CA3 | |

## 39.5 Directory of All Signal and Voltage Pins - Sorted by Pin Number

*Note:* The information in this chapter is preliminary and is subject to change.

### Table 39-6. Directory of Signal and Voltage Pins - Sorted by Pin Number

| | |
|---|---|
| A5 | VSS |
| A7 | VSS |
| A9 | VSS |
| A11 | DDR0_DQS_DP[9] |
| A13 | DDR0_DQ[4] |
| A15 | DDR1_ECC[2] |
| A18 | DDR1_DQS_DN[8] |
| A20 | DDR1_ECC[0] |
| A22 | VSS |
| A24 | DDR1_DQ[30] |
| A26 | DDR1_DQS_DP[12] |
| A28 | DDR1_DQ[28] |
| A30 | DDR1_DQ[2] |
| A32 | DDR1_DQS_DN[0] |
| A34 | DDR1_DQ[0] |
| A37 | VSS |
| A39 | CLK_X2_PAD |
| A41 | CLK_OUT_DN[0] |
| A43 | CLK_OUT_DN[1] |
| A45 | CLK_OUT_DN[2] |
| A48 | CLK_OUT_DN[3] |
| A50 | CLK_OUT_DN[4] |
| A52 | HSIO_RX_DP[0] |
| A54 | HSIO_RX_DP[1] |
| A57 | HSIO_RX_DP[2] |
| A59 | HSIO_RX_DP[3] |
| A61 | HSIO_RX_DP[4] |
| A63 | HSIO_RX_DP[5] |
| A65 | VSS |
| A66 | VSS |
| A68 | VSS |
| A70 | NC_ROT |
| B10 | DDR0_DQS_DN[0] |
| B12 | DDR0_DQ[0] |
| B14 | VSS |
| B16 | DDR1_ECC[6] |
| B19 | DDR1_DQS_DP[17] |
| B21 | DDR1_ECC[4] |
| B23 | DDR1_DQ[26] |
| B25 | DDR1_DQS_DN[3] |
| B27 | DDR1_DQ[24] |
| B29 | VSS |
| B31 | DDR1_DQ[6] |
| B33 | DDR1_DQS_DP[9] |
| B35 | DDR1_DQ[4] |
| B38 | VSS |
| B40 | VSS |
| B42 | VSS |
| B44 | VSS |
| B47 | VSS |
| B49 | VSS |
| B51 | VSS |
| B53 | VSS |
| B55 | VSS |
| B58 | VSS |
| B60 | VSS |
| B62 | VSS |
| B64 | VSS |
| C3 | VSS |
| C5 | VSS |
| C6 | DDR0_DQ[7] |
| C9 | DDR0_DQ[2] |
| C11 | DDR0_DQS_DN[9] |
| C13 | DDR0_DQ[5] |
| C15 | DDR1_ECC[3] |
| C18 | DDR1_DQS_DP[8] |
| C20 | DDR1_ECC[1] |
| C22 | VSS |
| C24 | DDR1_DQ[31] |
| C26 | DDR1_DQS_DN[12] |
| C28 | DDR1_DQ[29] |
| C30 | DDR1_DQ[3] |
| C32 | DDR1_DQS_DP[0] |
| C34 | DDR1_DQ[1] |
| C37 | VSS |
| C39 | CLK_X1_PAD |
| C41 | CLK_OUT_DP[0] |
| C43 | CLK_OUT_DP[1] |
| C45 | CLK_OUT_DP[2] |
| C48 | CLK_OUT_DP[3] |
| C50 | CLK_OUT_DP[4] |
| C52 | HSIO_RX_DN[0] |
| C54 | HSIO_RX_DN[1] |
| C57 | HSIO_RX_DN[2] |
| C59 | HSIO_RX_DN[3] |
| C61 | HSIO_RX_DN[4] |
| C63 | HSIO_RX_DN[5] |
| C65 | HSIO_RX_DN[6] |
| C67 | HSIO_RX_DP[6] |
| C68 | VSS |
| C70 | VSS |
| C72 | VSS |
| D8 | DDR0_DQ[6] |
| D10 | DDR0_DQS_DP[0] |
| D12 | DDR0_DQ[1] |
| D14 | VSS |
| D16 | DDR1_ECC[7] |
| D19 | DDR1_DQS_DN[17] |
| D21 | DDR1_ECC[5] |
| D23 | DDR1_DQ[27] |
| D25 | DDR1_DQS_DP[3] |
| D27 | DDR1_DQ[25] |
| D29 | VSS |
| D31 | DDR1_DQ[7] |
| D33 | DDR1_DQS_DN[9] |
| D35 | DDR1_DQ[5] |
| D38 | VSS |
| D40 | VSS |
| D42 | VSS |
| D44 | VSS |
| D47 | VSS |
| D49 | VSS |
| D51 | VSS |
| D53 | VSS |

| | |
|---|---|
| D55 | VSS |
| D58 | VSS |
| D60 | VSS |
| D62 | VSS |
| D64 | VSS |
| D66 | VSS |
| E1 | VSS |
| E3 | VSS |
| E6 | DDR0_DQ[3] |
| E9 | VSS |
| E11 | VSS |
| E13 | VSS |
| E15 | VSS |
| E18 | VSS |
| E20 | VSS |
| E22 | VSS |
| E24 | VSS |
| E26 | VSS |
| E28 | VSS |
| E30 | VSS |
| E32 | VSS |
| E34 | VSS |
| E37 | VSS |
| E39 | VSS |
| E41 | VSS |
| E43 | VSS |
| E45 | VSS |
| E48 | VSS |
| E50 | VSS |
| E52 | VSS |
| E54 | VSS |
| E57 | VSS |
| E59 | VSS |
| E61 | VSS |
| E63 | VSS |
| E65 | VSS |
| E67 | VSS |
| E70 | VSS |
| E72 | VSS |
| F3 | VSS |
| F5 | VSS |
| F8 | VSS |
| F66 | VSS |

| | |
|---|---|
| F69 | HSIO_RX_DN[7] |
| F70 | VSS |
| G1 | VSS |
| G6 | VSS |
| G7 | DDR1_BG0_BA[2] |
| G11 | DDR1_MA[12] |
| G15 | DDR1_CKE[3] |
| G18 | DDR1_DQ[18] |
| G22 | DDR1_DQS_DN[2] |
| G26 | DDR1_DQ[16] |
| G29 | VSS |
| G33 | LAN1_PORT0_TX_DN |
| G37 | LAN1_PORT1_TX_DN |
| G40 | LAN0_PORT0_TX_DN |
| G44 | LAN0_PORT1_TX_DN |
| G48 | HSIO_TX_DN[0] |
| G52 | HSIO_TX_DP[3] |
| G56 | HSIO_TX_DN[4] |
| G59 | HSIO_TX_DP[7] |
| G63 | HSIO_TX_DP[8] |
| G66 | VSS |
| G67 | HSIO_RX_DP[7] |
| G72 | VSS |
| H4 | VSS |
| H9 | DDR1_ACT_N_MA[15] |
| H13 | DDR1_CKE[2] |
| H17 | VSS |
| H20 | DDR1_DQ[22] |
| H24 | DDR1_DQS_DP[11] |
| H27 | DDR1_DQ[20] |
| H31 | VSS |
| H35 | RSVD_H35 |
| H38 | VSS |
| H42 | RSVD_H42 |
| H46 | VSS |
| H50 | VSS |
| H54 | HSIO_TX_DN[3] |
| H58 | VSS |
| H61 | HSIO_TX_DN[7] |
| H65 | HSIO_TX_DP[9] |
| H69 | HSIO_RX_DN[8] |
| J1 | VSS |
| J3 | VSS |

| | |
|---|---|
| J5 | DDR1_BG1_MA[14] |
| J7 | DDR1_MA[11] |
| J11 | DDR1_MA[9] |
| J15 | DDR1_CKE[1] |
| J18 | DDR1_DQ[19] |
| J22 | DDR1_DQS_DP[2] |
| J26 | DDR1_DQ[17] |
| J29 | VSS |
| J33 | LAN1_PORT0_TX_DP |
| J37 | LAN1_PORT1_TX_DP |
| J40 | LAN0_PORT0_TX_DP |
| J44 | LAN0_PORT1_TX_DP |
| J48 | HSIO_TX_DP[0] |
| J52 | HSIO_TX_DN[2] |
| J56 | HSIO_TX_DP[4] |
| J59 | HSIO_TX_DN[6] |
| J63 | HSIO_TX_DN[8] |
| J66 | HSIO_TX_DN[9] |
| J68 | VSS |
| J70 | HSIO_RX_DP[8] |
| J72 | VSS |
| K2 | VSS |
| K4 | VSS |
| K9 | VCCDDR_VDDQ |
| K13 | DDR1_CKE[0] |
| K17 | VSS |
| K20 | DDR1_DQ[23] |
| K24 | DDR1_DQS_DN[11] |
| K27 | DDR1_DQ[21] |
| K31 | VSS |
| K35 | RSVD_K35 |
| K38 | VSS |
| K42 | RSVD_K42 |
| K46 | VSS |
| K50 | HSIO_TX_DN[1] |
| K54 | VSS |
| K58 | HSIO_TX_DN[5] |
| K61 | VSS |
| K65 | HSIO_TX_DP[10] |
| K69 | VSS |
| K71 | VSS |
| L1 | DDR0_DQ[12] |
| L3 | DDR0_DQ[13] |

| | |
|---|---|
| L5 | VSS |
| L68 | VSS |
| L70 | HSIO_RX_DN[9] |
| L72 | HSIO_RX_DP[9] |
| M2 | DDR0_DQ[8] |
| M4 | DDR0_DQ[9] |
| M7 | DDR1_MA[6] |
| M11 | VCCDDR_VDDQ |
| M15 | DDR1_ALERT_N_PAR_ERR_N |
| M18 | VSS |
| M22 | VSS |
| M26 | VSS |
| M29 | VSS |
| M33 | VSS |
| M37 | VSS |
| M40 | VSS |
| M44 | VSS |
| M48 | HSIO_TX_DP[1] |
| M52 | HSIO_TX_DP[2] |
| M56 | HSIO_TX_DP[5] |
| M59 | HSIO_TX_DP[6] |
| M63 | HSIO_TX_DN[10] |
| M66 | VSS |
| M69 | VSS |
| M71 | VSS |
| N1 | DDR0_DQS_DP[10] |
| N3 | DDR0_DQS_DN[10] |
| N5 | VSS |
| N9 | DDR1_MA[7] |
| N13 | DDR1_MA[5] |
| N17 | VSS |
| N20 | VSS |
| N24 | VSS |
| N27 | VSS |
| N31 | VSS |
| N35 | VSS |
| N38 | VSS |
| N42 | VSS |
| N46 | VSS |
| N50 | VSS |
| N54 | VSS |
| N58 | VSS |

| | |
|---|---|
| N61 | VSS |
| N65 | HSIO_TX_DP[11] |
| N68 | VSS |
| N70 | HSIO_RX_DN[10] |
| N72 | HSIO_RX_DP[10] |
| P2 | DDR0_DQS_DN[1] |
| P4 | DDR0_DQS_DP[1] |
| P7 | DDR1_MA[3] |
| P11 | VCCDDR_VDDQ |
| P15 | VCCDDR_VDDQ |
| P18 | DDR1_DQ[10] |
| P22 | DDR1_DQS_DN[1] |
| P26 | DDR1_DQ[8] |
| P29 | VSS |
| P33 | RSVD_P33 |
| P37 | LAN1_PORT0_RX_DP |
| P40 | LAN1_PORT1_RX_DN |
| P44 | VSS |
| P48 | VSS |
| P52 | VSS |
| P56 | VSS |
| P59 | VSS |
| P63 | VSS |
| P66 | HSIO_TX_DN[11] |
| P69 | VSS |
| P71 | VSS |
| R1 | DDR0_DQ[14] |
| R3 | DDR0_DQ[15] |
| R5 | VSS |
| R9 | DDR1_MA[8] |
| R13 | DDR1_MA[4] |
| R17 | VSS |
| R20 | DDR1_DQ[14] |
| R24 | DDR1_DQS_DP[10] |
| R27 | DDR1_DQ[12] |
| R31 | RSVD_R31 |
| R35 | VSS |
| R38 | VSS |
| R42 | LAN0_PORT0_RX_DP |
| R46 | LAN0_PORT1_RX_DN |
| R50 | RSVD_R50 |
| R54 | RSVD_R54 |
| R58 | RSVD_R58 |

| | |
|---|---|
| R61 | VSS |
| R65 | HSIO_TX_DP[12] |
| R68 | VSS |
| R70 | HSIO_RX_DN[11] |
| R72 | HSIO_RX_DP[11] |
| T2 | DDR0_DQ[10] |
| T4 | DDR0_DQ[11] |
| T7 | DDR1_MA[2] |
| T11 | VCCDDR_VDDQ |
| T15 | DDR1_CLK_DP[3] |
| T18 | DDR1_DQ[11] |
| T22 | DDR1_DQS_DP[1] |
| T26 | DDR1_DQ[9] |
| T29 | VSS |
| T33 | RSVD_T33 |
| T37 | LAN1_PORT0_RX_DN |
| T40 | LAN1_PORT1_RX_DP |
| T44 | VSS |
| T48 | VSS |
| T52 | VSS |
| T56 | RSVD_T56 |
| T59 | RSVD_T59 |
| T63 | HSIO_TX_DN[12] |
| T66 | VSS |
| T69 | VSS |
| T71 | VSS |
| U9 | DDR1_MA[1] |
| U13 | DDR1_CLK_DN[3] |
| U17 | VSS |
| U20 | DDR1_DQ[15] |
| U24 | DDR1_DQS_DN[10] |
| U27 | DDR1_DQ[13] |
| U31 | RSVD_U31 |
| U35 | VSS |
| U38 | VSS |
| U42 | LAN0_PORT0_RX_DN |
| U46 | LAN0_PORT1_RX_DP |
| U50 | VSS |
| U54 | VSS |
| U58 | VSS |
| U61 | VSS |
| U65 | HSIO_TX_DP[13] |
| V1 | VSS |

| | |
|---|---|
| V3 | VSS |
| V5 | VSS |
| V7 | DDR1_PAR |
| V11 | VCCDDR_VDDQ |
| V15 | DDR1_CLK_DN[2] |
| V18 | VSS |
| V22 | VSS |
| V26 | VSS |
| V29 | VSS |
| V33 | VSS |
| V37 | VSS |
| V40 | VSS |
| V44 | VSS |
| V48 | VSS |
| V52 | VSS |
| V56 | VSS |
| V59 | VSS |
| V63 | VSS |
| V66 | HSIO_TX_DN[13] |
| V68 | VSS |
| V70 | HSIO_RX_DN[12] |
| V72 | HSIO_RX_DP[12] |
| W2 | DDR0_DQ[20] |
| W4 | DDR0_DQ[21] |
| W9 | VSS |
| W13 | VSS |
| W17 | DDR1_CLK_DP[2] |
| W20 | VSS |
| W24 | VSS |
| W50 | RSVD_W50 |
| W54 | ICLKRCOMP |
| W58 | LAN0_PORT1_SDP[3] |
| W61 | VSS |
| W65 | HSIO_TX_DP[14] |
| W69 | VSS |
| W71 | VSS |
| Y1 | DDR0_DQ[16] |
| Y3 | DDR0_DQ[17] |
| Y5 | VSS |
| Y7 | VSS |
| Y11 | VSS |
| Y15 | DDR1_CLK_DN[1] |
| Y18 | VCCDDR_VDDQ |
| Y22 | VSS |
| Y26 | VSS |
| Y27 | VSS |
| Y29 | VSS |
| Y30 | VSS |
| Y32 | VSS |
| Y34 | VSS |
| Y36 | OBS_VCCCPUVIDSI0GT_MOD0 |
| Y37 | VSS |
| Y39 | LAN1_RBIAS |
| Y41 | VSS |
| Y43 | VCCKR_HV |
| Y45 | VCCKR_HV |
| Y46 | VSS |
| Y48 | RSVD_Y48 |
| Y52 | VSS |
| Y56 | VSS |
| Y59 | SRTCRST_N |
| Y63 | HSIO_TX_DN[14] |
| Y66 | VSS |
| Y68 | VSS |
| Y70 | HSIO_RX_DN[13] |
| Y72 | HSIO_RX_DP[13] |
| AA2 | DDR0_DQS_DP[11] |
| AA4 | DDR0_DQS_DN[11] |
| AA9 | DDR0_DQ[28] |
| AA13 | VSS |
| AA17 | DDR1_CLK_DP[1] |
| AA20 | DDR1_BA[1] |
| AA24 | VSS |
| AA26 | VCCADDR_1P05 |
| AA27 | VCCACKDDR_1P05 |
| AA29 | RSVD_AA29 |
| AA30 | VCCREF_SFRXXXSI0_N |
| AA32 | VCCCPU_SVID_S0 |
| AA34 | VCCCPU_SVID_S0 |
| AA36 | OBS_VCCBIUVIDSI0GT_MOD0 |
| AA37 | OBS_VCCRAMCPUSI1GT_MOD0 |
| AA39 | VSS |
| AA41 | VCCKRLCPLL |
| AA43 | VCCKRLCPLL |
| AA45 | VCCKRLCPLL |
| AA46 | VCCKRLCPLL |
| AA48 | VSS |
| AA50 | LAN0_RBIAS |
| AA54 | VSS |
| AA58 | VSS |
| AA61 | VSS |
| AA65 | HSIO_TX_DP[15] |
| AA69 | VSS |
| AA71 | VSS |
| AB1 | DDR0_DQS_DN[2] |
| AB3 | DDR0_DQS_DP[2] |
| AB5 | VSS |
| AB68 | VSS |
| AB70 | HSIO_RX_DN[14] |
| AB72 | HSIO_RX_DP[14] |
| AC2 | DDR0_DQ[22] |
| AC4 | DDR0_DQ[23] |
| AC7 | DDR0_DQ[25] |
| AC11 | DDR0_DQ[24] |
| AC15 | DDR1_CLK_DN[0] |
| AC18 | VCCDDR_VDDQ |
| AC22 | DDR1_MA16_RAS_N |
| AC26 | VCCADDR_1P05 |
| AC27 | VCCACKDDR_1P05 |
| AC29 | VSS |
| AC30 | VCCRAM_S0 |
| AC32 | VCCCPU_SVID_S0 |
| AC34 | VCCCPU_SVID_S0 |
| AC36 | VCCRAM_S0 |
| AC37 | VSS |
| AC39 | VCCKRA_LV |
| AC41 | VCCKRA_LV |
| AC43 | VCCKRA_LV |
| AC45 | VCCKRA_LV |
| AC46 | VCCKRA_LV |
| AC48 | VCCKRA_LV |
| AC52 | VSS |
| AC56 | HSIO_COMPREF_DN |
| AC59 | LAN0_PORT0_SDP[2] |
| AC63 | VSS |
| AC66 | HSIO_TX_DN[15] |
| AC69 | VSS |

| | |
|---|---|
| AC71 | VSS |
| AD1 | DDR0_DQ[18] |
| AD3 | DDR0_DQ[19] |
| AD5 | VSS |
| AD9 | DDR0_DQ[29] |
| AD13 | VSS |
| AD17 | DDR1_CLK_DP[0] |
| AD20 | DDR1_BA[0] |
| AD24 | DDR01_COMP |
| AD50 | VSS |
| AD54 | HSIO_COMPREF_DP |
| AD58 | RSVD_AD58 |
| AD61 | VSS |
| AD65 | HSIO_TX_DP[16] |
| AD68 | VSS |
| AD70 | HSIO_RX_DN[15] |
| AD72 | HSIO_RX_DP[15] |
| AE2 | VSS |
| AE4 | VSS |
| AE7 | DDR0_DQS_DN[3] |
| AE11 | DDR0_DQS_DN[12] |
| AE15 | DDR1_ODT[2] |
| AE18 | VCCDDR_VDDQ |
| AE22 | DDR1_MA[0] |
| AE26 | VCCADDR_1P05 |
| AE27 | VCCACKDDR_1P05 |
| AE29 | VSS |
| AE30 | VCCRAM_S0 |
| AE32 | VCCCPU_SVID_S0 |
| AE34 | VCCCPU_SVID_S0 |
| AE36 | VCCRAM_S0 |
| AE37 | SENSE_VCCCPUVIDSI0 |
| AE39 | SENSE_VSSCPUVIDSI0 |
| AE41 | VCCKRA_LV |
| AE43 | VCCKRA_LV |
| AE45 | VCCKRA_LV |
| AE46 | VCCKRA_LV |
| AE48 | VSS |
| AE52 | VSS |
| AE56 | VSS |
| AE59 | RSMRST_N |
| AE63 | HSIO_TX_DN[16] |
| AE66 | VSS |

| | |
|---|---|
| AE69 | VSS |
| AE71 | VSS |
| AF1 | DDR0_ECC[4] |
| AF3 | DDR0_ECC[5] |
| AF5 | VSS |
| AF9 | DDR0_DQS_DP[12] |
| AF13 | VSS |
| AF17 | DDR1_CS_N[2] |
| AF20 | DDR1_MA[10] |
| AF24 | VSS |
| AF50 | VCCAON_HSIO_1P05 |
| AF54 | VSS |
| AF58 | VSS |
| AF61 | VSS |
| AF65 | HSIO_TX_DP[17] |
| AF68 | VSS |
| AF70 | HSIO_RX_DN[16] |
| AF72 | HSIO_RX_DP[16] |
| AG7 | DDR0_DQS_DP[3] |
| AG11 | DDR0_DQ[31] |
| AG15 | DDR1_MA17 |
| AG18 | VCCDDR_VDDQ |
| AG22 | DDR1_CS_N[0] |
| AG26 | VCCADDR_1P05 |
| AG27 | VCCACKDDR_1P05 |
| AG29 | VSS |
| AG30 | VCCRAM_S0 |
| AG32 | VCCCPU_SVID_S0 |
| AG34 | VCCCPU_SVID_S0 |
| AG36 | VCCRAM_S0 |
| AG37 | VSS |
| AG39 | VNN_SVID |
| AG41 | VSS |
| AG43 | VSS |
| AG45 | VCCDIGICKSI0_1P05 |
| AG46 | VSS |
| AG48 | VCCPLLEBB_HSIO_1P05 |
| AG52 | VCCAON_HSIO_1P05 |
| AG56 | COREPWROK |
| AG59 | RTC_X1_PAD |
| AG63 | VSS |
| AG66 | HSIO_TX_DN[17] |
| AH2 | DDR0_ECC[0] |

| | |
|---|---|
| AH4 | DDR0_ECC[1] |
| AH9 | DDR0_DQ[30] |
| AH13 | VSS |
| AH17 | DDR1_CS_N[3] |
| AH20 | RSVD_AH20 |
| AH24 | VCCDDRSFR_VDDQ |
| AH50 | VCCAON_HSIO_1P05 |
| AH54 | RSVD_AH54 |
| AH58 | RTC_X2_PAD |
| AH61 | VSS |
| AH65 | HSIO_TX_DP[18] |
| AH69 | VSS |
| AH71 | VSS |
| AJ1 | DDR0_DQS_DP[17] |
| AJ3 | DDR0_DQS_DN[17] |
| AJ5 | VSS |
| AJ7 | DDR0_DQ[26] |
| AJ11 | DDR0_DQ[27] |
| AJ15 | DDR1_C0 |
| AJ18 | VCCDDR_VDDQ |
| AJ22 | RSVD_AJ22 |
| AJ26 | VCCADDR_1P05 |
| AJ27 | VCCACKDDR_1P05 |
| AJ29 | VSS |
| AJ30 | VCCRAM_S0 |
| AJ32 | VCCCPU_SVID_S0 |
| AJ34 | VCCCPU_SVID_S0 |
| AJ36 | VCCRAM_S0 |
| AJ37 | VSS |
| AJ39 | VNN_SVID |
| AJ41 | VNN_SVID |
| AJ43 | VSS |
| AJ45 | VCCDIGICKSI0_1P05 |
| AJ46 | VSS |
| AJ48 | VCCCL_HSIO_1P05 |
| AJ52 | VSS |
| AJ56 | RTEST_N |
| AJ59 | INTRUDER_N |
| AJ63 | HSIO_TX_DN[18] |
| AJ66 | VSS |
| AJ68 | VSS |
| AJ70 | HSIO_RX_DN[17] |
| AJ72 | HSIO_RX_DP[17] |

| | |
|---|---|
| AK2 | DDR0_DQS_DN[8] |
| AK4 | DDR0_DQS_DP[8] |
| AK9 | VSS |
| AK13 | VSS |
| AK17 | DDR1_ODT[3] |
| AK20 | RSVD_AK20 |
| AK24 | VCCDDRSFR_VDDQ |
| AK26 | VCCADDR_1P05 |
| AK27 | VCCACKDDR_1P05 |
| AK29 | VSS |
| AK30 | VCCRAM_S0 |
| AK32 | VCCCPU_SVID_S0 |
| AK34 | VCCCPU_SVID_S0 |
| AK36 | VCCRAM_S0 |
| AK37 | VSS |
| AK39 | VNN_SVID |
| AK41 | VNN_SVID |
| AK43 | VSS |
| AK45 | VCCDIGXXXSI0_1P05 |
| AK46 | VSS |
| AK48 | VCCCL_HSIO_1P05 |
| AK50 | VCCAON_HSIO_1P05 |
| AK54 | VSS |
| AK58 | VSS |
| AK61 | VSS |
| AK65 | HSIO_TX_DP[19] |
| AK69 | VSS |
| AK71 | VSS |
| AL1 | DDR0_ECC[6] |
| AL3 | DDR0_ECC[7] |
| AL5 | VSS |
| AL7 | VSS |
| AL11 | VSS |
| AL15 | VSS |
| AL18 | VCCDDR_VDDQ |
| AL22 | RSVD_AL22 |
| AL52 | VSS |
| AL56 | NCSI_TXD[1] |
| AL59 | RSVD_AL59 |
| AL63 | VSS |
| AL66 | HSIO_TX_DN[19] |
| AL68 | VSS |
| AL70 | HSIO_RX_DN[18] |
| AL72 | HSIO_RX_DP[18] |
| AM2 | DDR0_ECC[2] |
| AM4 | DDR0_ECC[3] |
| AM26 | VCCADDR_1P05 |
| AM27 | VCCACKDDR_1P05 |
| AM29 | VSS |
| AM30 | VCCRAM_S0 |
| AM32 | VCCCPU_SVID_S0 |
| AM34 | VCCCPU_SVID_S0 |
| AM36 | VCCRAM_S0 |
| AM37 | VSS |
| AM39 | VNN_SVID |
| AM41 | VNN_SVID |
| AM43 | VSS |
| AM45 | VCCDIGXXXSI0_1P05 |
| AM46 | VSS |
| AM48 | VSS |
| AM69 | VSS |
| AM71 | VSS |
| AN1 | VSS |
| AN3 | VSS |
| AN5 | VSS |
| AN9 | DDR1_DQ[36] |
| AN13 | VSS |
| AN17 | VSS |
| AN20 | RSVD_AN20 |
| AN24 | VSS |
| AN50 | VCCAON_HSIO_1P05 |
| AN54 | USB2_COMP |
| AN58 | VSS |
| AN61 | USB2_DN[3] |
| AN65 | VSS |
| AN68 | VSS |
| AN70 | HSIO_RX_DN[19] |
| AN72 | HSIO_RX_DP[19] |
| AP2 | DDR01_DRAMRST_N |
| AP4 | DDR0_CKE[1] |
| AP7 | DDR1_DQ[33] |
| AP11 | DDR1_DQ[32] |
| AP15 | DDR1_DQ[45] |
| AP18 | VSS |
| AP22 | RSVD_AP22 |
| AP26 | VCCADDR_1P05 |
| AP27 | VCCACKDDR_1P05 |
| AP29 | VCCREF_DTS_E |
| AP30 | VCCRAM_S0 |
| AP32 | VCCCPU_SVID_S0 |
| AP34 | VCCCPU_SVID_S0 |
| AP36 | VCCRAM_S0 |
| AP37 | VSS |
| AP39 | VNN_SVID |
| AP41 | VNN_SVID |
| AP43 | VSS |
| AP45 | VCCDIGXXXSI0_1P05 |
| AP46 | VSS |
| AP48 | VCCAON_HSIO_1P05 |
| AP52 | VSS |
| AP56 | PCIE_CLKREQ_N[0] |
| AP59 | USB2_DP[3] |
| AP63 | USB2_DN[1] |
| AP66 | VSS |
| AP69 | VSS |
| AP71 | VSS |
| AR1 | VCCDDR_VDDQ |
| AR3 | DDR0_CKE[3] |
| AR5 | VSS |
| AR9 | DDR1_DQ[37] |
| AR13 | VSS |
| AR17 | DDR1_DQ[44] |
| AR20 | DDR1_MA14_WE_N |
| AR24 | VCCCKDDR_VDDQ |
| AR50 | VCCAON_HSIO_1P05 |
| AR54 | VCCUSB_1P05 |
| AR58 | VSS |
| AR61 | VSS |
| AR65 | USB2_DP[1] |
| AR68 | VSS |
| AR70 | USB2_DP[0] |
| AR72 | USB2_DN[0] |
| AT2 | DDR0_CKE[2] |
| AT4 | DDR0_CKE[0] |
| AT7 | DDR1_DQS_DN[4] |
| AT11 | DDR1_DQS_DN[13] |
| AT15 | DDR1_DQ[40] |
| AT18 | VSS |
| AT22 | DDR1_MA15_CAS_N |

| | |
|---|---|
| AT26 | VCCADDR_1P05 |
| AT27 | VCCACKDDR_1P05 |
| AT29 | VSS |
| AT30 | VCCRAM_S0 |
| AT32 | VCCCPU_SVID_S0 |
| AT34 | VCCCPU_SVID_S0 |
| AT36 | VCCRAM_S0 |
| AT37 | VSS |
| AT39 | VNN_SVID |
| AT41 | VNN_SVID |
| AT43 | VSS |
| AT45 | VCCDIGXXXSI0_1P05 |
| AT46 | VSS |
| AT48 | VCCAON_HSIO_1P05 |
| AT52 | VCCUSB_1P24 |
| AT56 | VSS |
| AT59 | VSS |
| AT63 | VSS |
| AT66 | VSS |
| AT69 | VSS |
| AT71 | VSS |
| AU9 | DDR1_DQS_DP[13] |
| AU13 | DDR1_DQS_DN[14] |
| AU17 | DDR1_DQ[41] |
| AU20 | DDR1_ODT[0] |
| AU24 | VCCCKDDR_VDDQ |
| AU26 | VCCADDR_1P05 |
| AU27 | VCCACKDDR_1P05 |
| AU29 | VSS |
| AU30 | VCCRAM_S0 |
| AU32 | VCCCPU_SVID_S0 |
| AU34 | VCCCPU_SVID_S0 |
| AU36 | VCCRAM_S0 |
| AU37 | VSS |
| AU39 | VCCFHVSI0_1P05 |
| AU41 | VNN_SVID |
| AU43 | VSS |
| AU45 | VCCREF_DTS_W |
| AU46 | VSS |
| AU48 | VCCRTC_3P3_G3 |
| AU50 | VCCUSBSUS_1P8 |
| AU54 | VCCUSB_1P05 |
| AU58 | NCSI_ARB_IN |

| | |
|---|---|
| AU61 | BVCCRTC_EXTPAD |
| AU65 | LAN0_PORT1_SDP[2] |
| AV1 | VCCDDR_VDDQ |
| AV3 | DDR0_ACT_N_MA[15] |
| AV5 | VSS |
| AV7 | DDR1_DQS_DP[4] |
| AV11 | DDR1_DQ[39] |
| AV15 | DDR1_DQS_DP[5] |
| AV18 | VSS |
| AV22 | DDR1_CS_N[1] |
| AV52 | VCCUSB_1P24 |
| AV56 | LAN0_PORT0_I2C_DATA |
| AV59 | LAN0_PORT0_I2C_CLK |
| AV63 | NCSI_ARB_OUT |
| AV66 | LAN0_PORT0_SDP[0] |
| AV68 | VSS |
| AV70 | USB2_DP[2] |
| AV72 | USB2_DN[2] |
| AW2 | DDR0_BG1_MA[14] |
| AW4 | DDR0_BG0_BA[2] |
| AW9 | DDR1_DQ[38] |
| AW13 | DDR1_DQS_DP[14] |
| AW17 | DDR1_DQS_DN[5] |
| AW20 | DDR1_MA[13] |
| AW24 | VSS |
| AW26 | VCCADDR_1P05 |
| AW27 | VCCACKDDR_1P05 |
| AW29 | VCCREF_SFRXXXSI0_S |
| AW30 | RSVD_AW30 |
| AW32 | VCCCPU_SVID_S0 |
| AW34 | VCCCPU_SVID_S0 |
| AW36 | SENSE_VCCRAMCPU_S0 |
| AW37 | SENSE_VSSRAMCPU_S0 |
| AW39 | VCCFHVSI0_GROUP1_1P05 |
| AW41 | VCCFHVSI0_GROUP2_1P05 |
| AW43 | VCCPADXXXSI0_1P8 |
| AW45 | VSS |
| AW46 | VCCPADXXXSI0_3P3 |
| AW48 | VCCUSBSUS_3P3 |
| AW50 | VCCUSBSUS_1P8 |
| AW54 | VSS |
| AW58 | VSS |

| | |
|---|---|
| AW61 | VSS |
| AW65 | VSS |
| AW69 | VSS |
| AW71 | VSS |
| AY1 | VCCDDR_VDDQ |
| AY3 | DDR0_ALERT_N_PAR_ERR_N |
| AY5 | VSS |
| AY7 | DDR1_DQ[34] |
| AY11 | DDR1_DQ[35] |
| AY15 | DDR1_DQ[47] |
| AY18 | VSS |
| AY22 | DDR1_ODT[1] |
| AY52 | DFX_PORT[9] |
| AY56 | DFX_PORT_CLK[0] |
| AY59 | NCSI_RXD[0] |
| AY63 | NCSI_CLK_IN |
| AY66 | LAN0_PORT1_SDP[0] |
| AY68 | VSS |
| AY70 | LAN1_PORT0_LED[0] |
| AY72 | LAN1_PORT0_LED[1] |
| BA2 | DDR0_MA[9] |
| BA4 | DDR0_MA[12] |
| BA9 | VSS |
| BA13 | VSS |
| BA17 | DDR1_DQ[46] |
| BA20 | VSS |
| BA24 | RSVD_BA24 |
| BA26 | RSVD_BA26 |
| BA27 | VSS |
| BA29 | RSVD_BA29 |
| BA30 | VSS |
| BA32 | VCCCPU_SVID_S0 |
| BA34 | VCCCPU_SVID_S0 |
| BA36 | OBS_VCCCPUVIDSI0GT_MOD4 |
| BA37 | VSS |
| BA39 | VCCFHVIFPSI0_1P8 |
| BA41 | VCCFHVIFPSI0_3P3 |
| BA43 | VCCPADXXXSI0_1P8 |
| BA45 | VSS |
| BA46 | VCCPADXXXSI0_3P3 |
| BA48 | VCCUSBSUS_3P3 |
| BA50 | VSS |

| | |
|---|---|
| BA54 | SVID_ALERT_N |
| BA58 | PCIE_CLKREQ_N[1] |
| BA61 | THERMTRIP_N |
| BA65 | LAN0_PORT1_LED[0] |
| BA69 | LAN0_PORT0_SDP[3] |
| BA71 | LAN0_PORT0_SDP[1] |
| BB1 | DDR0_MA[7] |
| BB3 | DDR0_MA[11] |
| BB5 | VSS |
| BB7 | VSS |
| BB11 | VSS |
| BB15 | DDR1_DQ[43] |
| BB18 | VSS |
| BB22 | VSS |
| BB26 | VSS |
| BB29 | RSVD_BB29 |
| BB33 | VCCCPU_SVID_S0 |
| BB37 | VSS |
| BB40 | VSS |
| BB44 | VSS |
| BB48 | VSS |
| BB52 | VSS |
| BB56 | VSS |
| BB59 | VSS |
| BB63 | VSS |
| BB66 | LAN0_PORT1_I2C_CLK |
| BB68 | VSS |
| BB70 | DFX_PORT_CLK[1] |
| BB72 | LAN0_PORT1_SDP[1] |
| BC2 | DDR0_MA[6] |
| BC4 | DDR0_MA[8] |
| BC69 | LAN0_PORT1_I2C_DATA |
| BC71 | VSS |
| BD1 | VCCDDR_VDDQ |
| BD3 | DDR0_MA[5] |
| BD5 | VSS |
| BD9 | VSS |
| BD13 | VSS |
| BD17 | DDR1_DQ[42] |
| BD20 | VSS |
| BD24 | VSS |
| BD27 | VSS |
| BD31 | VSS |
| BD35 | VCCCPU_SVID_S0 |
| BD38 | VCCFHVSI0_GROUP0_1P05 |
| BD42 | HSHV_IRCOMP |
| BD46 | VCC_LPC_ESPI_3P3_1P8 |
| BD50 | ESPI_IRCOMP |
| BD54 | USB_OC_N |
| BD58 | ESPI_IO[0] |
| BD61 | PCIE_CLKREQ_N[4] |
| BD65 | NCSI_CRS_DV |
| BD68 | VSS |
| BD70 | NCSI_TX_EN |
| BD72 | NCSI_RXD[1] |
| BE2 | DDR0_MA[3] |
| BE4 | DDR0_MA[4] |
| BE7 | VSS |
| BE11 | DDR1_DQS_DN[15] |
| BE15 | VSS |
| BE18 | VSS |
| BE22 | DDR0_DQ[41] |
| BE26 | DDR0_DQS_DP[5] |
| BE29 | DDR0_DQ[43] |
| BE33 | VCCCPU_SVID_S0 |
| BE37 | VSS |
| BE40 | EMMC_IRCOMP |
| BE44 | CTBTRIGINOUT |
| BE48 | DFX_PORT[13] |
| BE52 | GPIO[6] |
| BE56 | SVID_DATA |
| BE59 | SVID_CLK |
| BE63 | DFX_PORT[2] |
| BE66 | NCSI_TXD[0] |
| BE69 | LAN0_PORT0_LED[0] |
| BE71 | VSS |
| BF1 | DDR0_MA[2] |
| BF3 | DDR0_MA[1] |
| BF5 | VSS |
| BF9 | DDR1_DQ[52] |
| BF13 | VSS |
| BF17 | VSS |
| BF20 | DDR0_DQ[45] |
| BF24 | DDR0_DQS_DN[14] |
| BF27 | DDR0_DQ[47] |
| BF31 | VSS |
| BF35 | VCCCPU_SVID_S0 |
| BF38 | RSVD_BF38 |
| BF42 | VSS |
| BF46 | VSS |
| BF50 | VSS |
| BF54 | VSS |
| BF58 | VSS |
| BF61 | VSS |
| BF65 | VSS |
| BF68 | VSS |
| BF70 | GPIO[0] |
| BF72 | LAN0_PORT0_LED[1] |
| BG7 | DDR1_DQ[53] |
| BG11 | DDR1_DQS_DP[15] |
| BG15 | DDR1_DQ[60] |
| BG18 | VSS |
| BG22 | DDR0_DQ[40] |
| BG26 | DDR0_DQS_DN[5] |
| BG29 | DDR0_DQ[42] |
| BG33 | VCCCPU_SVID_S0 |
| BG37 | VSS |
| BG40 | RSVD_BG40 |
| BG44 | TDI |
| BG48 | FLEX_CLK_SE[0] |
| BG52 | SMB_ME_SMT0_ALRT_N |
| BG56 | SPI_CS_N[0] |
| BG59 | PMU_SUSCLK |
| BG63 | DFX_PORT[6] |
| BG66 | LAN0_PORT1_LED[1] |
| BH2 | DDR0_CLK_DP[3] |
| BH4 | DDR0_CLK_DN[3] |
| BH9 | DDR1_DQ[49] |
| BH13 | DDR1_DQ[61] |
| BH17 | DDR1_DQ[56] |
| BH20 | DDR0_DQ[44] |
| BH24 | DDR0_DQS_DP[14] |
| BH27 | DDR0_DQ[46] |
| BH31 | VSS |
| BH35 | VCCCPU_SVID_S0 |
| BH38 | RSVD_BH38 |
| BH42 | CX_PREQ_N |
| BH46 | GPIO[7] |

| | |
|---|---|
| BH50 | GPIO[4] |
| BH54 | SMB_LAN_CLK |
| BH58 | SUSPWRDNACK |
| BH61 | SPI_IO[2] |
| BH65 | ESPI_IO[3] |
| BH69 | DFX_PORT[5] |
| BH71 | PCIE_CLKREQ_N[2] |
| BJ1 | VCCDDR_VDDQ |
| BJ3 | VCCDDR_VDDQ |
| BJ5 | VSS |
| BJ7 | DDR1_DQ[48] |
| BJ11 | VSS |
| BJ15 | DDR1_DQS_DN[16] |
| BJ18 | VSS |
| BJ22 | VSS |
| BJ26 | VSS |
| BJ29 | VSS |
| BJ33 | VCCCPU_SVID_S0 |
| BJ37 | VSS |
| BJ40 | VSS |
| BJ44 | VSS |
| BJ48 | VSS |
| BJ52 | VSS |
| BJ56 | VSS |
| BJ59 | VSS |
| BJ63 | VSS |
| BJ66 | PCIE_CLKREQ_N[3] |
| BJ68 | VSS |
| BJ70 | VSS |
| BJ72 | ESPI_CS_N[0] |
| BK2 | DDR0_CLK_DP[1] |
| BK4 | DDR0_CLK_DN[1] |
| BK9 | DDR1_DQS_DP[6] |
| BK13 | DDR1_DQS_DP[16] |
| BK17 | DDR1_DQ[57] |
| BK20 | VSS |
| BK24 | VSS |
| BK27 | VSS |
| BK31 | VSS |
| BK35 | VCCCPU_SVID_S0 |
| BK38 | RSVD_BK38 |
| BK42 | DFX_PORT[11] |
| BK46 | EMMC_CLK |

| | |
|---|---|
| BK50 | SMB_HOST_DATA |
| BK54 | SMB_LAN_DATA |
| BK58 | PCIE_CLKREQ_N[7] |
| BK61 | ESPI_ALRT_N[0] |
| BK65 | PMU_SLP_S3_N |
| BK69 | GPIO[1] |
| BK71 | GPIO[2] |
| BL1 | DDR0_CLK_DP[2] |
| BL3 | DDR0_CLK_DN[2] |
| BL5 | VSS |
| BL7 | DDR1_DQS_DN[6] |
| BL11 | DDR1_DQ[50] |
| BL15 | DDR1_DQS_DP[7] |
| BL18 | VSS |
| BL22 | DDR0_DQ[57] |
| BL26 | DDR0_DQS_DP[7] |
| BL29 | DDR0_DQ[59] |
| BL33 | VCCCPU_SVID_S0 |
| BL37 | VSS |
| BL40 | TRST_N |
| BL44 | EMMC_STROBE |
| BL48 | GPIO[5] |
| BL52 | ERROR_N[0] |
| BL56 | UART0_RXD |
| BL59 | PMU_RESETBUTTON_N |
| BL63 | PMU_PLTRST_N |
| BL66 | SPI_CS_N[1] |
| BL68 | VSS |
| BL70 | MEMHOT_N |
| BL72 | VSS |
| BM2 | DDR0_CLK_DP[0] |
| BM4 | DDR0_CLK_DN[0] |
| BM9 | DDR1_DQ[55] |
| BM13 | DDR1_DQS_DN[7] |
| BM17 | DDR1_DQ[58] |
| BM20 | DDR0_DQ[61] |
| BM24 | DDR0_DQS_DN[16] |
| BM27 | DDR0_DQ[63] |
| BM31 | VSS |
| BM35 | VCCCPU_SVID_S0 |
| BM38 | GPIO[150] |
| BM42 | VSS |
| BM46 | VSS |

| | |
|---|---|
| BM50 | VSS |
| BM54 | VSS |
| BM58 | VSS |
| BM61 | VSS |
| BM65 | VSS |
| BM69 | VSS |
| BM71 | PROCHOT_N |
| BN1 | VDDQ |
| BN3 | DDR0_PAR |
| BN5 | VSS |
| BN7 | DDR1_DQ[54] |
| BN11 | VSS |
| BN15 | DDR1_DQ[63] |
| BN18 | VSS |
| BN22 | DDR0_DQ[56] |
| BN26 | DDR0_DQS_DN[7] |
| BN29 | DDR0_DQ[58] |
| BN33 | VCCCPU_SVID_S0 |
| BN37 | VSS |
| BN40 | TCK |
| BN44 | GPIO[3] |
| BN48 | EMMC_CMD |
| BN52 | MCERR_N |
| BN56 | GPIO[12] |
| BN59 | UART1_TXD |
| BN63 | SATA_PDETECT[1] |
| BN66 | SMB_LAN_ALRT_N |
| BN68 | DFX_PORT[0] |
| BN70 | DFX_PORT[1] |
| BN72 | VSS |
| BP4 | DDR0_MA[0] |
| BP69 | DFX_PORT[3] |
| BR1 | VSS |
| BR3 | VSS |
| BR6 | VSS |
| BR9 | DDR1_DQ[51] |
| BR13 | DDR1_DQ[62] |
| BR17 | DDR1_DQ[59] |
| BR20 | DDR0_DQ[60] |
| BR24 | DDR0_DQS_DP[16] |
| BR27 | DDR0_DQ[62] |
| BR31 | VSS |
| BR35 | VCCCPU_SVID_S0 |

| | |
|---|---|
| BR38 | VSS |
| BR42 | DFX_PORT[10] |
| BR46 | SMB_ME_SMT0_CLK |
| BR50 | SMB_HOST_CLK |
| BR54 | SMB_PECI_CLK |
| BR58 | ERROR_N[1] |
| BR61 | PMU_WAKE_N |
| BR65 | SATA0_LED_N |
| BR67 | ESPI_IO[1] |
| BR70 | VSS |
| BR72 | VSS |
| BT5 | DDR0_BA[1] |
| BT8 | VSS |
| BT66 | VSS |
| BT69 | DFX_PORT[7] |
| BU1 | VSS |
| BU3 | VSS |
| BU6 | DDR0_MA[10] |
| BU9 | VSS |
| BU11 | VSS |
| BU13 | VSS |
| BU15 | VSS |
| BU18 | VSS |
| BU20 | VSS |
| BU22 | VSS |
| BU24 | VSS |
| BU26 | VSS |
| BU28 | VSS |
| BU30 | VSS |
| BU32 | VCCCPU_SVID_S0 |
| BU34 | VCCCPU_SVID_S0 |
| BU37 | VSS |
| BU39 | VSS |
| BU41 | VSS |
| BU43 | EMMC_D[4] |
| BU45 | VSS |
| BU48 | VSS |
| BU50 | VSS |
| BU52 | VSS |
| BU54 | VSS |
| BU57 | SATA0_SDOUT |
| BU59 | VSS |
| BU61 | VSS |
| BU63 | SPI_MOSI_IO[0] |
| BU65 | VSS |
| BU67 | VSS |
| BU70 | VSS |
| BU72 | VSS |
| BV8 | DDR0_BA[0] |
| BV10 | DDR0_CS_N[2] |
| BV12 | DDR0_CS_N[1] |
| BV14 | DDR0_C0 |
| BV16 | VSS |
| BV19 | DDR0_DQ[33] |
| BV21 | DDR0_DQS_DP[4] |
| BV23 | DDR0_DQ[35] |
| BV25 | DDR0_DQ[53] |
| BV27 | DDR0_DQS_DN[15] |
| BV29 | DDR0_DQ[55] |
| BV31 | VSS |
| BV33 | VCCCPU_SVID_S0 |
| BV35 | VSS |
| BV38 | TDO |
| BV40 | DFX_PORT[15] |
| BV42 | EMMC_D[7] |
| BV44 | EMMC_D[1] |
| BV47 | SMB_ME_SMT0_DATA |
| BV49 | SMB_LEG_DATA |
| BV51 | PCIE_CLKREQ_N[6] |
| BV53 | UART1_RXD |
| BV55 | SATA1_LED_N |
| BV58 | SUS_STAT_N |
| BV60 | ADR_TRIGGER |
| BV62 | SPI_MISO_IO[1] |
| BV64 | ESPI_IO[2] |
| BV66 | ESPI_RST_N |
| BW1 | VSS |
| BW3 | VSS |
| BW5 | DDR0_CS_N[0] |
| BW6 | DDR0_MA16_RAS_N |
| BW9 | DDR0_MA14_WE_N |
| BW11 | DDR0_MA[13] |
| BW13 | DDR0_ODT[2] |
| BW15 | DDR0_CS_N[3] |
| BW18 | DDR0_DQ[37] |
| BW20 | DDR0_DQS_DN[13] |
| BW22 | DDR0_DQ[39] |
| BW24 | VSS |
| BW26 | DDR0_DQ[49] |
| BW28 | DDR0_DQS_DP[6] |
| BW30 | DDR0_DQ[51] |
| BW32 | VCCCPU_SVID_S0 |
| BW34 | VCCCPU_SVID_S0 |
| BW37 | GPIO[151] |
| BW39 | TMS |
| BW41 | DFX_PORT[12] |
| BW43 | VSS |
| BW45 | EMMC_D[0] |
| BW48 | SMB_LEG_ALRT_N |
| BW50 | IERR_N |
| BW52 | PCIE_CLKREQ_N[5] |
| BW54 | SATA1_SDOUT |
| BW57 | VSS |
| BW59 | PMU_SLP_S45_N |
| BW61 | SPI_IO[3] |
| BW63 | VSS |
| BW65 | VSS |
| BW67 | DFX_PORT[8] |
| BW68 | DFX_PORT[4] |
| BW70 | VSS |
| BW72 | VSS |
| BY10 | DDR0_ODT[0] |
| BY12 | VCCDDR_VDDQ |
| BY14 | DDR0_ODT[1] |
| BY16 | VSS |
| BY19 | DDR0_DQ[32] |
| BY21 | DDR0_DQS_DN[4] |
| BY23 | DDR0_DQ[34] |
| BY25 | DDR0_DQ[52] |
| BY27 | DDR0_DQS_DP[15] |
| BY29 | DDR0_DQ[54] |
| BY31 | VSS |
| BY33 | VCCCPU_SVID_S0 |
| BY35 | VSS |
| BY38 | CX_PRDY_N |
| BY40 | VSS |
| BY42 | EMMC_D[6] |
| BY44 | EMMC_D[2] |
| BY47 | VSS |

| | |
|---|---|
| BY49 | SMB_PECI_DATA |
| BY51 | SMB_LEG_CLK |
| BY53 | VSS |
| BY55 | GPIO[8] |
| BY58 | RSVD_BY58 |
| BY60 | VSS |
| BY62 | ESPI_ALRT_N[1] |
| BY64 | ESPI_CLK |
| CA3 | VSS_ROT |
| CA5 | VSS |
| CA7 | VSS |
| CA9 | VDDQ |
| CA11 | DDR0_MA15_CAS_N |
| CA13 | DDR0_MA17 |
| CA15 | DDR0_ODT[3] |
| CA18 | DDR0_DQ[36] |
| CA20 | DDR0_DQS_DP[13] |
| CA22 | DDR0_DQ[38] |
| CA24 | VSS |
| CA26 | DDR0_DQ[48] |
| CA28 | DDR0_DQS_DN[6] |
| CA30 | DDR0_DQ[50] |
| CA32 | VCCCPU_SVID_S0 |
| CA34 | VCCCPU_SVID_S0 |
| CA37 | GPIO[152] |
| CA39 | CTBTRIGOUT |
| CA41 | DFX_PORT[14] |
| CA43 | EMMC_D[5] |
| CA45 | EMMC_D[3] |
| CA48 | FLEX_CLK_SE[1] |
| CA50 | ERROR_N[2] |
| CA52 | UART0_TXD |
| CA54 | GPIO[9] |
| CA57 | SATA_PDETECT[0] |
| CA59 | PMU_PWRBTN_N |
| CA61 | SPI_CLK |
| CA63 | ESPI_CS_N[1] |
| CA65 | VSS |
| CA66 | VSS |
| CA68 | VSS |
| CA70 | VSS |

§ §

# 40 Signal Electrical and Timing Characteristics

This chapter is organized by signal interface. Each sub-chapter contains the interface DC, AC, and signal timing requirements and characteristics. Some of these requirements and characteristics are based on industry and Intel standards. When the SoC interface complies to a standard, the reader is referred to the standard for the requirements and characteristics. SoC exceptions to the standard, if any, are shown as well as any of the standard's optional interface characteristics as implemented by the SoC design.

Most of the information presented here is in Table and Figure form. In some cases, the reader is referred to one of the functional-description chapters where additional interface-related information is available.

## 40.1 DDR Memory Interface

The SoC supports up to two channels of DDR4, depending on configuration and product SKU.

### 40.1.1 DDR4

DDR4 SDRAM is the successor to DDR3 SDRAM. It was revealed at the Intel Developer Forum in San Francisco in 2008, samples were announced in early 2011. DDR4 is expected to reach mass market adoption around 2015.

Published in September 2012, the initial JEDEC DDR4 standard has been defined to provide higher performance, with improved reliability and reduced power, thereby representing a significant achievement relative to previous DRAM memory technologies.

The SoC DDR4 interface was designed to the DDR4 (JESD79-4) standard, and is available for free download from the JEDEC website.

The SoC supports DDR4 voltage of 1.2V. The DDR4 standard supports various options, see the Memory Controller Section 9.3, "Feature List" on page 353 of this document for the DDR4 options supported by this SoC.

This SoC was designed to the DDR4 standard, aside from the deviations listed below.

#### 40.1.1.1 Deviations From The DDR4 Standard

While there are no known deviations, DDR4 is very sensitive to board layout, and only the configurations listed in the SoC Platform Design Guide have been simulated and validated by Intel. It is up to the customer to simulate and validate memory configurations not listed within the SoC Platform Design Guide.

#### 40.1.1.2 Supplementary DDR4 Signals Not Covered By The DDR4 Standard

No supplementary signals.

## 40.2 PCI Express Root Port Interface

The SoC has up to 16 PCI Express 3.0 lanes, depending on configuration and product SKU, see the PCI Express Section 11.2, "Feature List" on page 372 section of this document for the PCI Express options supported by this SoC.

The SoC PCI Express interface was designed to the PCI Express 3.0 Base specification revision 3.0, made available in November 2010. In August 2007, PCI-SIG announced that PCI Express 3.0 would carry a bit rate of 8 gigatransfers per second (GT/s), and that it would be backward compatible with existing PCI Express implementations. New features for the PCI Express 3.0 specification include a number of optimizations for enhanced signaling and data integrity, including transmitter and receiver equalization, PLL improvements, clock data recovery, and channel enhancements for currently supported topologies.

PCI Express 3.0 upgrades the encoding scheme to 128b/130b from the previous 8b/10b encoding, reducing the overhead to approximately 1.54% ((130–128)/130), as opposed to the 20% overhead of PCI Express 2.0. This is achieved by a technique called "scrambling" that applies a known binary polynomial to a data stream in a feedback topology. Because the scrambling polynomial is known, the data can be recovered by running it through a feedback topology using the inverse polynomial. PCI Express 3.0's 8 GT/s bit rate effectively delivers 985 MB/s per lane, practically doubling the lane bandwidth relative to PCI Express 2.0.

On November 18, 2010, the PCI Special Interest Group officially published the finalized PCI Express 3.0 specification to its members to build devices based on this new version of PCI Express.

PCI Express 3.1 was released in 2014, making various tweaks to the published standard.

### 40.2.1 Deviations From The PCI Express 3.0 Standard

No known deviations.

### 40.2.2 Supplementary PCI Express Signals Not Covered By The PCI Express 3.0 Standard

No supplementary signals.

## 40.3 SATA Interface

The SoC has up to 16 SATA 3.1 lanes, depending on configuration and product SKU, see Chapter 12, "SATA Controller" of this document for the SATA options supported by this SoC.

The Serial ATA International Organization presented the draft specification of SATA 6 Gbit/s physical layer in July 2008, and ratified its physical layer specification on August 18, 2008. The SoC SATA interface was designed to this standard.

The full 3.0 standard was released on May 27, 2009. It runs with a native transfer rate of 6.0 Gbit/s, and taking 8b/10b encoding into account, the maximum uncoded transfer rate is 4.8 Gbit/s (600 MB/s). The theoretical burst throughput of SATA 6.0 Gbit/s is double that of SATA 2.0. SATA 3.1 was released July 2011, introducing new features, supporting the same transfer rate as SATA 3.0, both being backward compatible with SATA 2.0.

### 40.3.1 Deviations From The SATA 3.1 Standard

No known deviations.

### 40.3.2 Supplementary SATA Signals Not Covered By The SATA 3.1 Standard

No supplementary signals.

## 40.4 USB Interface

The SoC has up to four USB 3.0 ports, depending on configuration and product SKU, see the USB Section 14.2, "Feature List" on page 495 of this document for the USB options supported by this SoC.

The SoC USB interface was designed to the USB 3.0 standard released in November 2008. The standard defines a new SuperSpeed mode with a signaling speed of 5 Gbit/s and, due to encoding overhead, usable data rate of up to 4 Gbit/s (500 MB/s). A USB 3.0 port is usually colored blue, and is backwards compatible with USB 2.0.

While the SoC does NOT support USB 3.1, the Universal Serial Bus Revision 3.1 Specification was released July, 2013. The USB 3.1 specification introduces a faster transfer mode called "SuperSpeed USB 10 Gbps"; its logo features a SuperSpeed+ (stylized as SUPERSPEED+) caption. The USB 3.1 standard increases the signaling rate to 10 Gbit/s, double that of USB 3.0, and reduces line encoding overhead to just 3% by changing the encoding scheme to 128b/132b.

The USB 3.1 standard is backward compatible with USB 3.0, and USB 3.0 is backward compatible with USB 2.0.

A USB 3.0 port makes use of SuperSpeed PHYs, in addition to USB 2.0 PHYs. The SoC USB controller supports using just the USB 2.0 PHYs, to implement USB 2.0 ports, or combine the USB 2.0 PHYs with the SuperSpeed PHYs to implement a fully compliant USB 3.0 port; up to a total of four USB ports, depending on configuration and product SKU.

### 40.4.1 Deviations From The USB 3.0 Standard

No known deviations.

### 40.4.2 Supplementary USB Signals Not Covered By The USB 3.0 Standard

No supplementary signals.

### 40.4.3 USB 2.0 Standard

Refer to the *Universal Serial Bus Specification, Revision 2.0*.

## 40.5 SMBus Interface

The SoC has several SMBus controllers, depending on configuration and product SKU, see the SMBus Section 27.2, "Feature List" on page 709 of this document for the SMBus options supported by this SoC.

The SoC SMBus interface is designed to the SMBus 3.0 standard.

### 40.5.1 Deviations From The SMBus 2.0 Standard

SMBus/I2C Characteristics of the SDA and SCL Bus Do Not Meet Specification with minor deviation:

SMBus/I2C tSU;STA (set-up time for a repeated START condition) and tSU;STO (set-up time for STOP condition) show certain level violation for 100KHz speed, 400KHz speed and 1MHz Speed.

- For 100KHz tsusta min spec 4.7us vs 4.62 measured, 1.59%
- For 100KHz tsusto min spec 4.0us vs 3.96 measured, 0.847%
- For 400KHz tsusta min spec 0.6us vs 0.529 measured, 11.8%
- For 1MHz tsusta min spec 0.26us vs 0.202 measured, 22%
- For 1MHz tsusto min spec 0.26us vs 0.240 measured, 7.58%

SMBus/I2C tHIght (High period of the SCL clock) shows certain level violation for 400KHz speed and 1MHz Speed.

- For 400KHz tHight min spec 0.6us vs 0.59us measured, 0.7%
- For 1MHz tHight min spec 0.26us vs 0.2us measured, 0.847%

### 40.5.2 Supplementary SMBus Signals Not Covered By The SMBus 2.0 Standard

No supplementary signals.

## 40.6 LPC Interface

The SoC LPC interface was designed to the LPC Specification 1.1, August 2000, see the LPC Section 20.2, "Feature List" on page 628 of this document for the LPC options supported by this SoC.

### 40.6.1 Deviations From The LPC 1.1 Standard

No known deviations.

### 40.6.2 Supplementary LPC Signals Not Covered By The LPC 1.1 Standard

No supplementary signals.

## 40.7 eMMC Interface

The SoC eMMC interface was designed to the JEDEC Standard Embedded Multi-Media Card (eMMC) Electrical Standard (5.0), JESD84-B50 (Revision of JESD84-B451, June 2012) September 2013, see the eMMC Section 17.2, "Feature List" on page 558 of this document for the eMMC options supported by this SoC.

The eMMC standard specifies the electrical characteristics at the device end, and not the host. The electrical characteristics at the device end are greatly affected by the physical routing between the host and device. See the SoC Platform Design Guide for routing guidelines that meet the eMMC electrical characteristics at the device end of the channel.

### 40.7.1 Deviations From The eMMC 5.0 Standard

Given the electrical characteristics of our eMMC I/Os, and the SoC routing guidelines provided by the SoC Platform Design Guide for the eMMC signals, smaller overshoot/undershoot excursions than what the eMMC standard allows will be seen at the device end of the channel. This does not mean we deviate from the eMMC standard, rather we more strictly control overshoot/undershoot than is required by the eMMC standard.

### 40.7.2 Supplementary eMMC Signals Not Covered By The eMMC 5.0 Standard

No supplementary signals.

## 40.8 HSUART Interface

The SoC HSUARTs are compatible with the Texas Instruments* PC16550D HSUART specification. See the HSUART Section 18.2, "Feature List" on page 568 of this document for the HSUART options supported by this SoC.

### 40.8.1 Deviations From Texas Instruments PC16550D Standard

The I/Os of the original Texas Instruments PC16550D used 5.0V signaling. The HSUARTs within the SoC deviate from the Texas Instruments PC16550D standard in that the SoC HSUART I/Os are implemented using the 3.3V GPIO buffers, see Section 40.13.3, "3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics" on page 1066 for the Electrical Characteristics.

### 40.8.2 Supplementary HSUART Signals Not Covered By The Texas Instruments PC16550D Standard

No supplementary signals.

## 40.9 Serial Peripheral Interface (SPI)

See Chapter 19, "Serial Peripheral Interface (SPI)" of this document for the SPI options supported by this SoC.

There is no industry standard for the SPI bus. The electrical and timing characteristics for this interface are described below.

### 40.9.1 Interface Timing Parameters and Waveforms

Table 40-1, Table 40-2, and Table 40-3 contain the timing specifications for the SPI interface signals.

**Table 40-1. SPI (17.14MHz) Signal Timing Specifications**

| Symbol | Parameter | Min | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| t180 | Serial Clock Frequency | 16.9 | - | 17.4 | MHz |
| t183 | $T_{CO}$ of SPI_MOSI_IO[0] with respect to serial clock falling edge at the host. | -5 | - | 13 | ns |
| t184 | Setup of SPI_MISO_IO[1] with respect to serial clock falling edge at the host. | 16 | - | - | ns |
| t185 | Hold of SPI_MISO _IO[1] with respect to serial clock falling edge at the host. | 0 | - | - | ns |
| t186 | Setup of SPI_CS_N[1:0] assertion with respect to serial clock rising at the host. | 30 | - | - | ns |
| t187 | Hold of SPI_CS_N[1:0] deassertion with respect to serial clock falling at the host. | 30 | - | - | ns |
| t188 | SPI_CLK High Time | 26.37 | - | - | ns |
| t189 | SPI_CLK Low Time | 26.82 | - | - | ns |

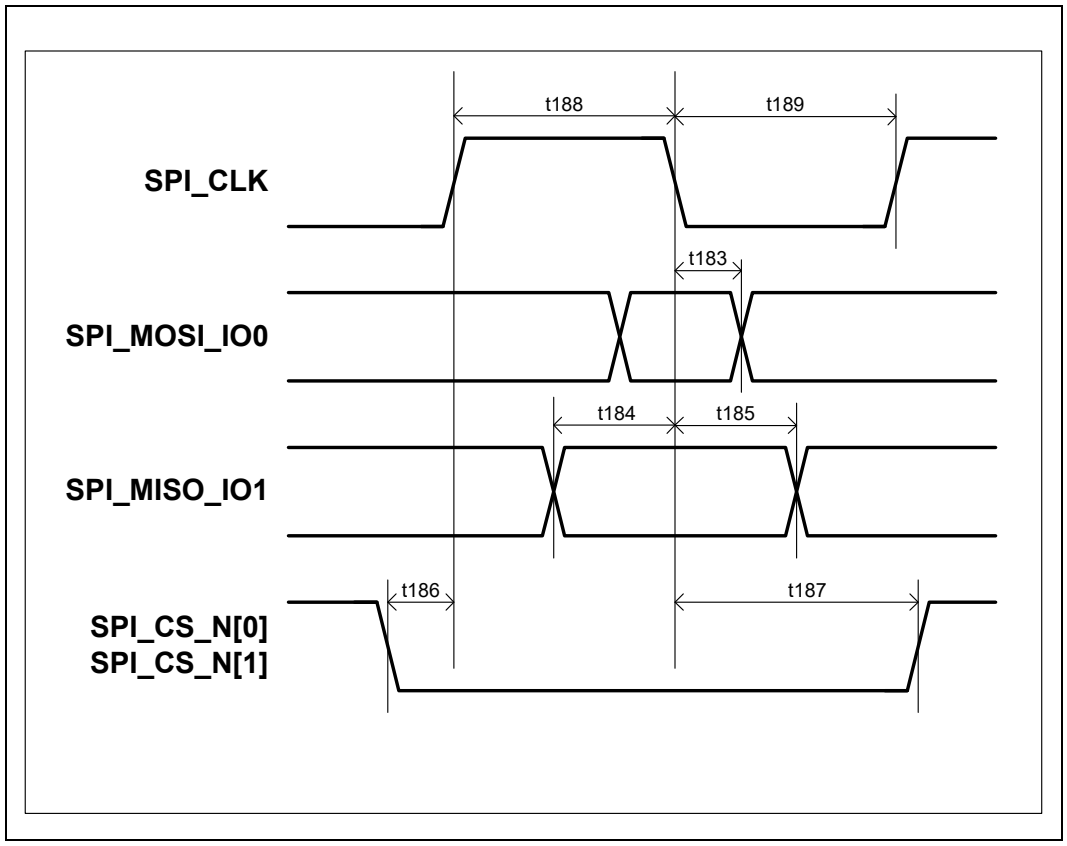
**Table 40-2. SPI (30MHz) Signal Timing Specifications**

| Symbol | Parameter | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| t180 | Serial Clock Frequency | 29.83 | - | 30.6 | MHz |
| t183 | $T_{CO}$ of SPI_MOSI _IO[0] with respect to serial clock falling edge at the host. | -5 | - | 5 | ns |
| t184 | Setup of SPI_MISO_IO[1] with respect to serial clock falling edge at the host. | 8 | - | - | ns |
| t185 | Hold of SPI_MISO_IO[1] with respect to serial clock falling edge at the host. | 0 | - | - | ns |
| t186 | Setup of SPI_CS_N[1:0] assertion with respect to serial clock rising at the host. | 30 | - | - | ns |
| t187 | Hold of SPI_CS_N[1:0] deassertion with respect to serial clock falling at the host. | 30 | - | - | ns |
| t188 | SPI_CLK High Time | 14.88 | - | - | ns |
| t189 | SPI_CLK Low Time | 15.18 | - | - | ns |

### Table 40-3. SPI (48MHz) Signal Timing Specifications

| Symbol | Parameter | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|---|
| t180 | Serial Clock Frequency | 47 | - | 49 | MHz |
| t183 | $T_{CO}$ of SPI_MOSI_IO[0] with respect to serial clock falling edge at the host. | -3 | - | 4.5 | ns |
| t184 | Setup of SPI_MISO_IO[1] with respect to serial clock falling edge at the host. | 5 | - | - | ns |
| t185 | Hold of SPI_MISO_IO[1] with respect to serial clock falling edge at the host. | 0 | - | - | ns |
| t186 | Setup of SPI_CS_N[1:0] assertion with respect to serial clock rising at the host. | 30 | - | - | ns |
| t187 | Hold of SPI_CS_N[1:0] deassertion with respect to serial clock falling at the host. | 30 | - | - | ns |
| t188 | SPI_CLK High Time | 5 | - | - | ns |
| t189 | SPI_CLK Low Time | 10.8 | - | - | ns |

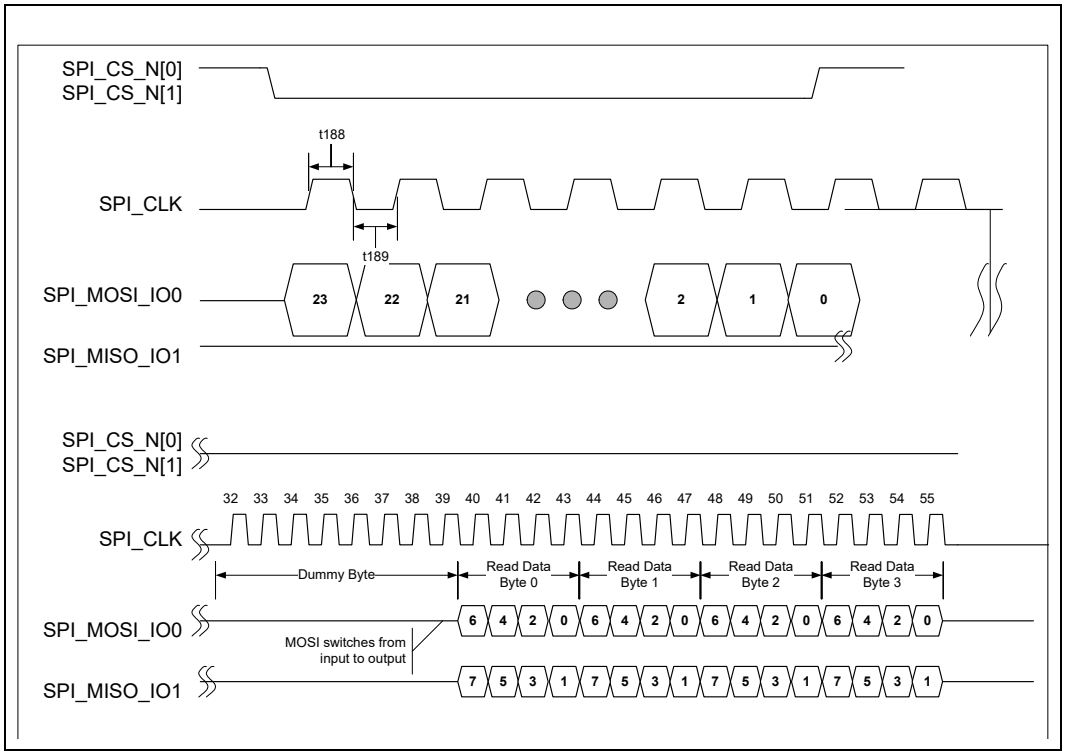
### Figure 40-1. SPI Timing Diagram



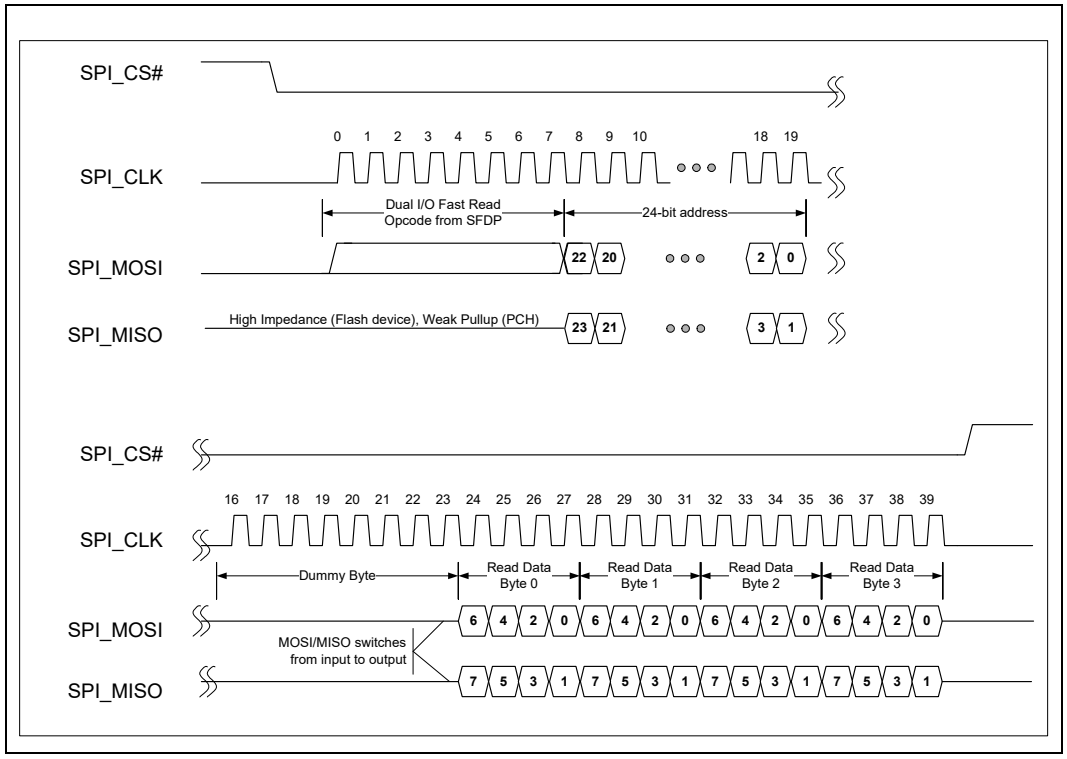
### Table 40-4. CS# Setup and CS# Hold

| Parameter | Value | Description |
|---|---|---|
| CS# Setup | 2.5 ns (min) | SPI_CS# low to SPI_CLK high |
| CS# Hold | 2.5 ns (min) | SPI_CLK low to SPI_CS# low |

### Figure 40-2. Dual-Output Fast Read Timing with 24-bit Addressing



## Figure 40-3. Dual I/O Fast Read Timing with 24-bit Addressing



## Figure 40-4. Quad Output Fast Read Timing with 24-bit Addressing

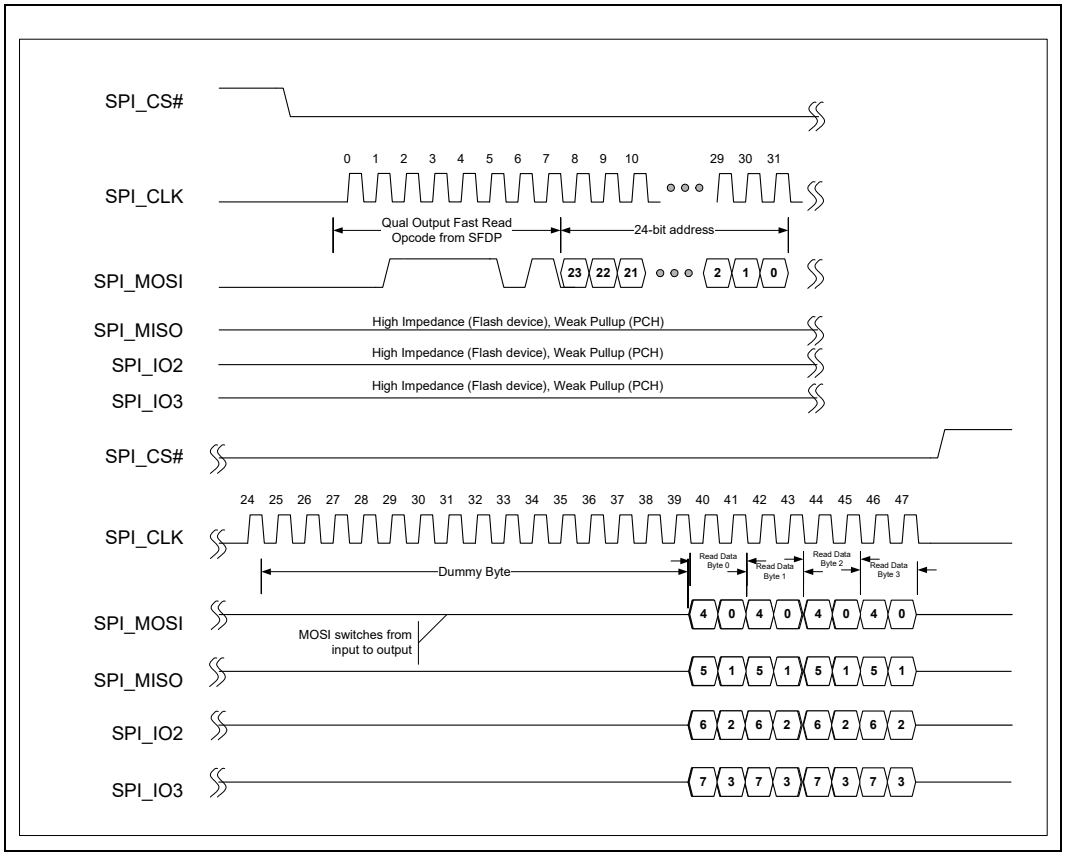
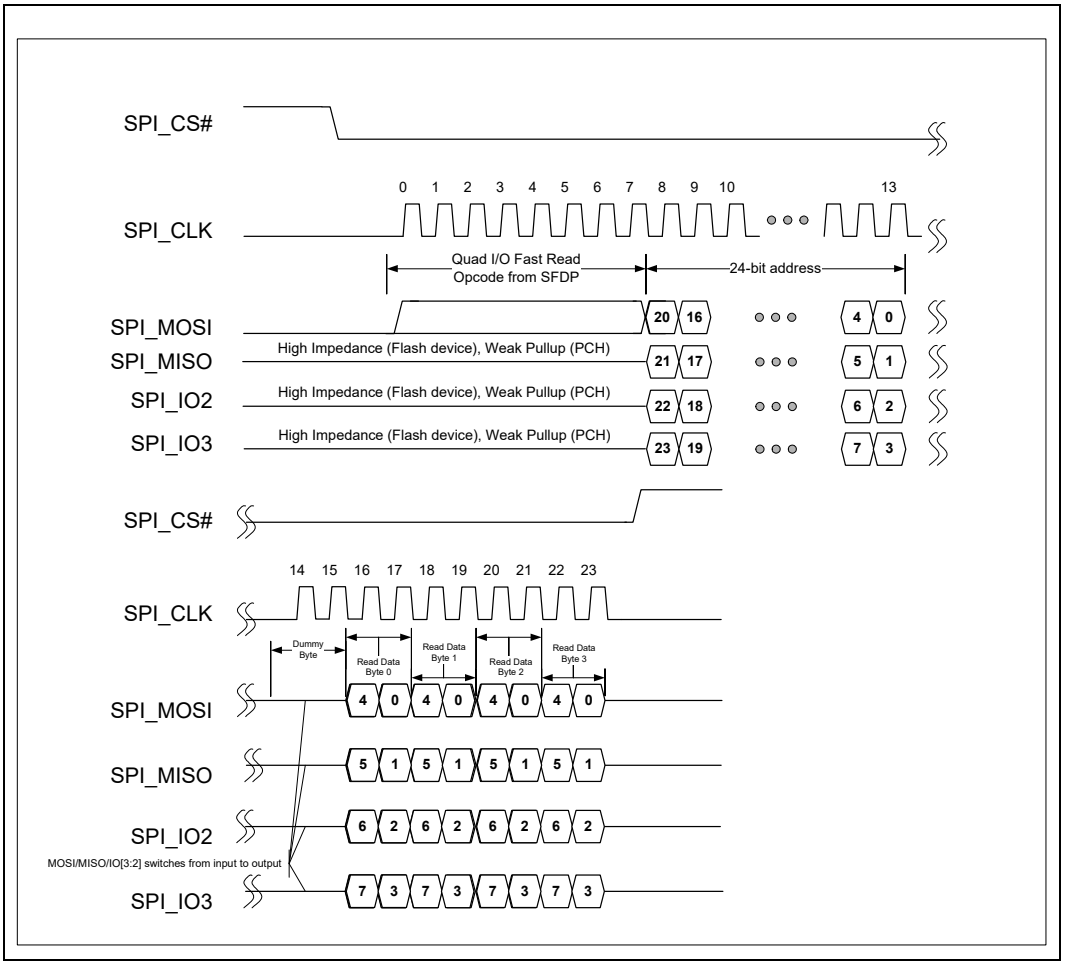


**Figure 40-5. Quad I/O Fast Read Timing with 24-bit Addressing**




## 40.9.2 SPI Electrical Characteristics

The I/O buffers used for the SPI interface are the 3.3V GPIO (CMOS-HV Buffer Type) buffers, see Section 40.13.3, "3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics" on page 1066 for the Electrical Characteristics.

## 40.10 LAN Interface

The SoC includes up to two integrated LAN controllers, with up to two LAN ports each, depending on configuration and product SKU. See the LAN Section 13.3, "Feature List" on page 417 of this document for the LAN options supported by this SoC.

Each LAN port is configurable, supporting various electrical interfaces, depending on configuration and product SKU. Each type of supported electrical interface is listed separately in the sections below.

### 40.10.1 10GBASE-KR Interface

The SoC 10GBASE-KR interface was designed to the IEEE802.3 Standard.

#### 40.10.1.1 Deviations From The IEEE802.3 10GBASE-KR Standard

No known deviations.

#### 40.10.1.2 Supplementary 10GBASE-KR Signals Not Covered By The IEEE802.3 10GBASE-KR Standard

No supplementary signals.

### 40.10.2 SFI Interface

The SoC SFI interface was designed to the SFF-8431 Revision 4.1 Standard.

#### 40.10.2.1 Deviations From The SFF-8431 SFI Standard

The SoC supports only SFP+ DA (Direct Attach; i.e., passive twin-ax cable assembly) and limiting optical modules. It will not support linear optical modules.

Linear modules are modules which contain a linear receiver, whereas limiting modules contain a limiting receiver.

#### 40.10.2.2 Supplementary SFI Signals Not Covered By The SFF-8431 SFI Standard

No supplementary signals.

### 40.10.3 2500BASE-X Interface

2500BASE-X is not an IEEE standard, but leverages the electricals from IEEE802.3ae-2002 (XAUI). 2500BASE-X is term used by Broadcom to describe 2.5Gbit/s (3.125GBd) operation.

#### 40.10.3.1 Deviations From The 2500BASE-X Industry Convention

No known deviations.

#### 40.10.3.2 Supplementary 2500BASE-X Signals Not Covered By The 2500BASE-X Industry Convention

No supplementary signals.

### 40.10.4 1000BASE-KX Interface

The SoC 1000BASE-KX interface was designed to the IEEE802.3 Standard.

#### 40.10.4.1 Deviations From The IEEE802.3 1000BASE-KX Standard

No known deviations.

#### 40.10.4.2 Supplementary 1000BASE-KX Signals Not Covered By The IEEE802.3 1000BASE-KX Standard

No supplementary signals.

### 40.10.5 SGMII Interface

The SoC SGMII interface was designed to the Cisco Systems* SGMII Industry Standard, Revision 1.7.

#### 40.10.5.1 Deviations From The Cisco Systems SGMII Industry Standard

No known deviations, other than Half Duplex mode is not supported, only Full Duplex mode is supported.

#### 40.10.5.2 Supplementary SGMII Signals Not Covered By The Cisco Systems SGMII Industry Standard

No supplementary signals.

### 40.10.6 Management Data Input/Output (MDIO)

The MDIO interface is also known as Serial Management Interface (SMI) or Media Independent Interface Management (MIIM). The MDIO interface connects Media Access Control (MAC) devices with Ethernet physical layer (PHY) circuits.

The MDIO interface is defined by Clause 22 of IEEE 802.3, and is augmented by Clause 45.

The SoC MDIO interface supports both Clause 22 and Clause 45. However, the only I/O voltage supported is the 3.3V option.

#### 40.10.6.1 Deviations From IEEE 802.3

Clause 22 and Clause 45 of IEEE 802.3 support multiple voltage options for the I/O buffers. The only I/O voltage option supported by the SoC is 3.3V, and the specific I/O buffers used for the MDIO interface are the 3.3V GPIO (CMOS-HV Buffer Type) buffers, see Section 40.13.3, "3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics" on page 1066 for the Electrical Characteristics.

#### 40.10.6.2 Supplementary MDIO Signals Not Covered By Clause 22 and Clause 45 of IEEE 802.3

No supplementary signals.

### 40.10.7 Slave ("Target") Network Controller Sideband Interface (NC-SI)

The SoC NC-SI Slave interface was designed to the *DMTF Network Controller Sideband Interface (NC-SI) Specification, Document Number: DSP0222, Version 1.0.0.*

#### 40.10.7.1 Deviations From The NC-SI Standard

No known deviations.

#### 40.10.7.2 Supplementary NC-SI Signals Not Covered By The NC-SI Standard

No supplementary signals.

## 40.11 SoC Serial Voltage Identification (SVID) Interface

It consists of three signals. See Section 34.6, "Serial Voltage Identification (sVID) Controller" on page 908 of this document for the options supported by this SoC.

### 40.11.1 SoC SVID Timing Deviates from the VR13.0 Specification

The SoC SVID timing deviates from the VR13.0 specification for the following parameters:

- Tco_max_CPU Clock to data delay @bump = 5µs versus the 0.65µs listed in the VR13.0 specification
- Tco_min_CPU Clock to data delay @bump = 1µs versus the -3.6µs listed in the VR13.0 specification
- Tsu_CPU - Setup time of signal VDIO at CPU side = 15µs versus the 1µs listed in the VR13.0 specification.

### 40.11.2 Supplementary Signals Not Covered By The SVID Standard

No supplementary signals.

## 40.12 SoC JTAG and Debug Interfaces

The SoC JTAG/Debug interface was designed to the IEEE Std 1149.1 Standard. See Chapter 38, "JTAG and Debug Ports" of this document for the options supported by this SoC.

The IEEE standard for JTAG does not specify the electrical characteristics of the interface signals, only the functional requirements. The intent is to not limit JTAG by the electrical characteristics of any existing device or I/O technology, but to support any device with existing and future I/O technologies. The JTAG specifications include guidance for interconnecting multiple devices using disparate I/O technologies.

As is the case, the I/O technology used to design the SoC JTAG/Debug I/Os are implemented using the 1.05V GPIO buffers, see Section 40.13.1, "1.05V GPIO (CMOS-LV Buffer Type) Electrical Characteristics" on page 1062 for the Electrical Characteristics.

### 40.12.1 Deviations From The JTAG Standard

No known deviations.

### 40.12.2 Supplementary Signals Not Covered By The JTAG Standard

No supplementary signals.

## 40.13 Customer General Purpose I/O (GPIO) Interface

The SoC has several pins that are configurable as GPIO. See the GPIO Section 26.1, "GPIO Ball and Signal Names" on page 686 of this document for the GPIO options supported by this SoC.

GPIO transceivers are Intel custom circuit designs, and do not comply to any public specifications. There are three classes (types) of GPIO transceivers used in this SoC: 1.05V GPIO, 1.8V GPIO, and 3.3V GPIO.

Many SoC signals make use of the same buffers as the GPIO signals. All SoC signals are listed in Table 39-1. SoC signals which are implemented using the GPIO buffers contain either CMOS-LV, CMOS-MV, or CMOS-HV in the column "Buffer Type" of Table 39-1.

The electrical characteristics for each class of GPIO is listed below in separate sections.

### 40.13.1 1.05V GPIO (CMOS-LV Buffer Type) Electrical Characteristics

#### 40.13.1.1 Receiver Electrical Characteristics

The tables below list the DC and AC Characteristics of the receiver.

**Table 40-5. 1.05V GPIO Receiver DC Characteristics**

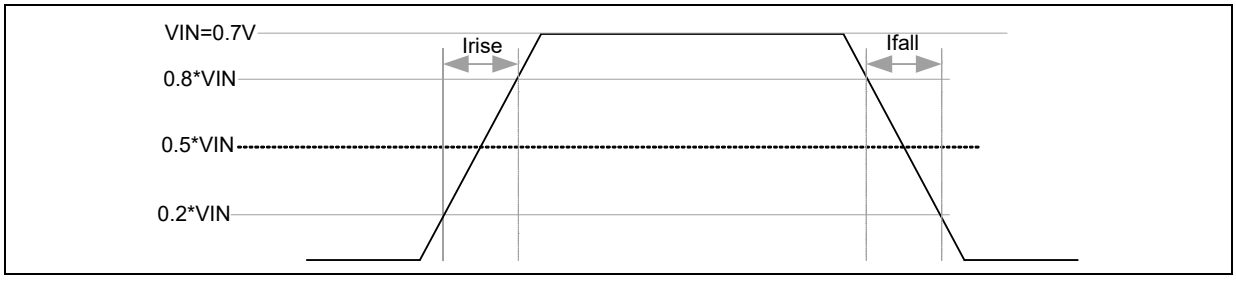
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V1P05 | | 1.05 | | N/A | V | +/- 5% |
| High-Level Input Voltage | Vih | 0.65*V1P05 | | | N/A | V | |
| Low-Level Input Voltage | Vil | | | 0.35*V1P05 | N/A | V | |
| Input Hysteresis | Vhys | 50 | | | N/A | mV | |
| Input Leakage | Lin | -50 | | 50 | V1P05/GND | uA | |
| Input Capacitance | Cin | | | 5 | N/A | pF | |

**Table 40-6. 1.05V GPIO Receiver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Unit | Notes |
|---|---|---|---|---|---|---|
| Input Frequency | Fin | N/A | | 533 | MHz | |
| Input Rise/Fall | Irise/Ifall | N/A | | 6 | ns | |

The figure below depicts the receiver timing.

**Figure 40-6. 1.05V GPIO Receiver Timing**



### 40.13.1.2 Driver Electrical Characteristics

The tables below list the DC and AC Characteristics of the driver.

**Table 40-7. 1.05V GPIO Driver DC Characteristics**

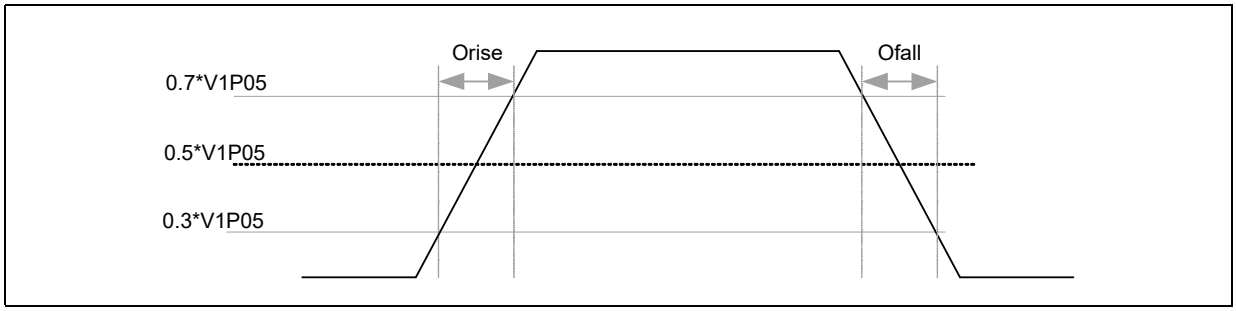
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V1P05 | | 1.05 | | N/A | V | +/- 5% |
| Driver Pullup Impedance | Zup | 30 | 50 | 90 | N/A | Ohm | |
| Driver Pulldown Impedance | Zdn | 12 | 20 | 28 | N/A | Ohm | |
| High-Level Output Voltage | Voh | V1P05-0.3 | | | -2mA | V | |
| Low-Level Output Voltage | Vol | | | 0.2 | 3mA | V | |

**Table 40-8. 1.05V GPIO Driver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition (Cap to GND) | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Output Frequency | Fout | N/A | | 533 | 0.15pF | MHz | |
| Output Rise / Fall Rate | Orise/Ofall | N/A | N/A | 0.2 | 0.15pF | ns | Output Rise Rate = -(Output Fall Rate) |

The figure below depicts the driver timing.

**Figure 40-7. 1.05V GPIO Driver Timing**



## 40.13.2 1.8V GPIO (CMOS-MV Buffer Type) Electrical Characteristics

### 40.13.2.1 Receiver Electrical Characteristics

The tables below list the DC and AC Characteristics of the receiver.

**Table 40-9. 1.8V GPIO Receiver DC Characteristics**

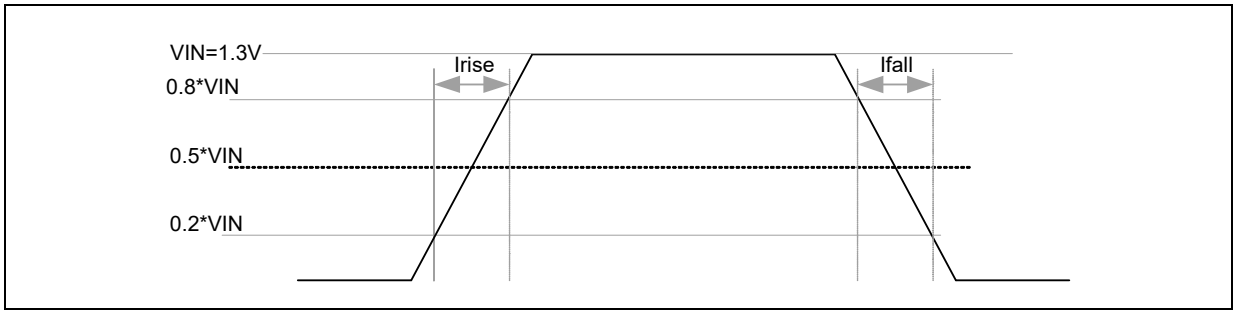
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V1P80 | | 1.8 | | N/A | V | +/- 5% |
| High-Level Input Voltage | Vih | 0.65*V1P80 | | | N/A | V | |
| Low-Level Input Voltage | Vil | | | 0.35*V1P80 | N/A | V | |
| Input Hysteresis | Vhys | 100 | | | N/A | mV | |
| Input Leakage | Lin | -10 | | 10 | V1P80/GND | uA | |
| Input Capacitance | Cin | | | 5 | N/A | pF | |

**Table 40-10. 1.8V GPIO Receiver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Unit | Notes |
|---|---|---|---|---|---|---|
| Input Frequency | Fin | N/A | | 400 | MHz | |
| Input Rise/Fall | Irise/Ifall | N/A | | 6 | ns | |

The figure below depicts the receiver timing.

**Figure 40-8. 1.8V GPIO Receiver Timing**



### 40.13.2.2 Driver Electrical Characteristics

The tables below list the DC and AC Characteristics of the driver.

**Table 40-11. 1.8V GPIO Driver DC Characteristics**

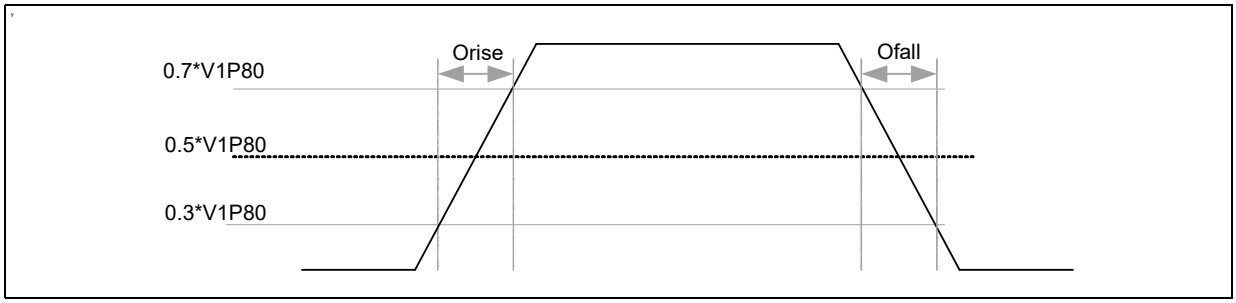
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V1P80 | | 1.8 | | N/A | V | +/- 5% |
| Output Impedance | Zout | 40 | 50 | 60 | N/A | Ohm | |
| High-Level Output Voltage | Voh | V1P80-0.45 | | | -2mA | V | |
| Low-Level Output Voltage | Vol | | | 0.45 | 2mA | V | |

**Table 40-12. 1.8V GPIO Driver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition (Cap to GND) | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Output Frequency | Fout | N/A | | 100<br>200 | 30pF<br>15pF | MHz | |
| Output Rise / Fall Rate | Orise/Ofall | 0.4 | N/A | 0.7 | 30pF | V /ns | Output Rise Rate = -(Output Fall Rate) |

The figure below depicts the driver timing.

**Table 40-13. 1.8V GPIO Driver Timing**



## 40.13.3 3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics

### 40.13.3.1 Receiver Electrical Characteristics

The tables below list the DC and AC Characteristics of the receiver.

**Table 40-14. 3.3V GPIO Receiver DC Characteristics**

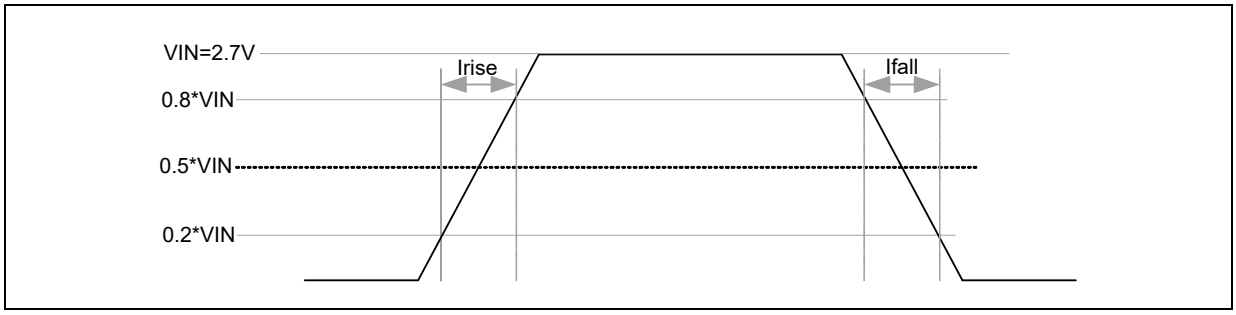
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V3P30 | | 3.3 | | N/A | V | +/- 5% |
| High-Level Input Voltage | Vih | 0.625*V3P30 | | | N/A | V | |
| Low-Level Input Voltage | Vil | | | 0.25*V3P30 | N/A | V | |
| Input Hysteresis | Vhys | 50 | | | N/A | mV | |
| Input Leakage | Lin | -10 | | 10 | V3P30/GND | uA | |
| Input Capacitance | Cin | | | 7 | N/A | pF | |

**Table 40-15. 3.3V GPIO Receiver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Unit | Notes |
|---|---|---|---|---|---|---|
| Input Frequency | Fin | N/A | | 208 | MHz | |
| Input Rise/Fall | Irise/Ifall | N/A | | 6 | ns | |

The figure below depicts the receiver timing.

**Table 40-16. 3.3V GPIO Receiver Timing**



### 40.13.3.2 Driver Electrical Characteristics

The tables below list the DC and AC Characteristics of the driver.

**Table 40-17. 3.3V GPIO Driver DC Characteristics**

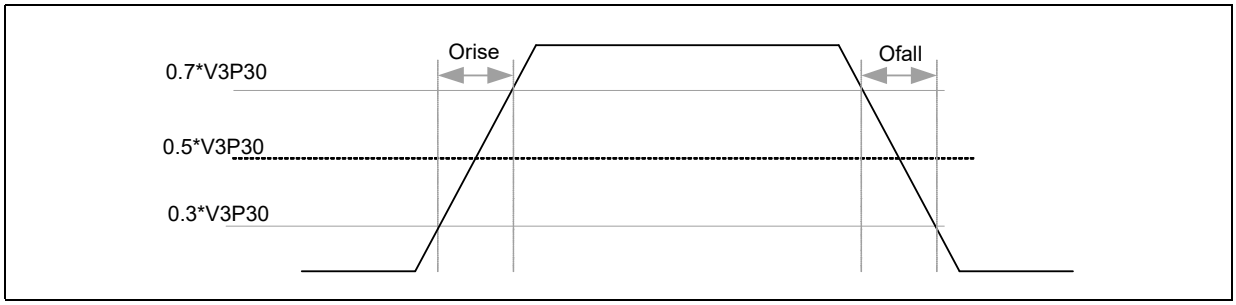
| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Power Supply | V3P30 | | 3.3 | | N/A | V | +/- 5% |
| Output Impedance | Zout | 40 | 50 | 60 | N/A | Ohm | |
| High-Level Output Voltage | Voh | V3P30-0.45 | | | -3mA | V | |
| Low-Level Output Voltage | Vol | | | 0.45 | 3mA | V | |

**Table 40-18. 3.3V GPIO Driver AC Characteristics**

| Parameter | Symbol | Minimum | Nominal | Maximum | Test Condition (Cap to GND) | Unit | Notes |
|---|---|---|---|---|---|---|---|
| Output Frequency | Fout | N/A | | 125<br>50 | 15pF<br>30pF | MHz | |
| Output Rise / Fall Rate | Orise/Ofall | 0.6 | | 1.4 | 30pF | V /ns | Output Rise Rate = -(Output Fall Rate) |

The figure below depicts the driver timing.

**Figure 40-9. 3.3V GPIO Driver Timing**



## 40.14 Miscellaneous Interfaces

Miscellaneous signals not included in the sections above, are not defined by any industry standard, and are custom SoC signals defined by Intel. Three different types of Intel custom electrical interfaces were used in implementing these interfaces.

Many SoC signals make use of the same buffers as the GPIO signals. All SoC signals are listed in Table 39-1, "Signal Pin Name Directory - Sorted by Name" on page 995. SoC signals which are implemented using the GPIO buffers contain either CMOS-LV, CMOS-MV, or CMOS-HV in the column "Buffer Type" of Table 39-1.

Chapter 39, "Signal Pin Names and Signal MUXing" lists all of the SoC signals along with their "V Group" (Voltage Group).

See the subsections below for the electrical characteristics of each "V Group".

### 40.14.1 Signals of "V Group V1P05" (CMOS-LV Buffer Type)

See Section 40.13.1, "1.05V GPIO (CMOS-LV Buffer Type) Electrical Characteristics" on page 1062 for the Electrical Characteristics of signals within this group.

### 40.14.2 Signals of "V Group V1P80" (CMOS-MV Buffer Type)

See Section 40.13.2, "1.8V GPIO (CMOS-MV Buffer Type) Electrical Characteristics" on page 1064 for the Electrical Characteristics of signals within this group.

### 40.14.3 Signals of "V Group V3P30" (CMOS-HV Buffer Type)

See Section 40.13.3, "3.3V GPIO (CMOS-HV Buffer Type) Electrical Characteristics" on page 1066 for the Electrical Characteristics of signals within this group.

§ §

# 41 Operating Conditions and Power Requirements

## 41.1 Absolute Maximum and Minimum Ratings

For proper functional operation, all processor electrical and thermal requirements must be satisfied. These are shown starting with Section 41.2, "Normal Operating Conditions" on page 1070.

When the device is subjected to conditions outside the functional operation condition limits, but within absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. If a device is returned to conditions within functional operation limits after having been subjected to conditions outside these limits, but within the absolute maximum and minimum ratings, the device may be functional, but with its lifetime degraded depending on exposure to conditions exceeding the functional operation condition limits.

At conditions exceeding absolute maximum and minimum ratings, neither functionality nor long-term reliability is expected. Moreover, devices that are subjected to these conditions for any length of time, will either not function or its reliability will be severely degraded when returned to conditions within the functional operating condition limits.

### 41.1.1 Component Storage Conditions Specification

This section applies to component-level storage prior to board attach. Environmental storage condition limits define temperature and relative humidity to which the device is exposed to while being stored in the applicable Intel shipping media trays, reels, and moisture barrier bags and boxes, and the component is not electrically connected.

#### 41.1.1.1 Prior to Board-Attach

Table 41-1 specifies absolute maximum and minimum storage temperature and humidity limits for given time durations. Failure to adhere to specified limits can result in physical damage to the component. If this is suspected, Intel recommends a visual inspection to determine possible physical damage to the silicon or surface components.

**Table 41-1. Storage Condition Ratings - Prior to Board-Attach**

| Symbol | Parameter | Minimum | Maximum | Unit |
|---|---|---|---|---|
| $T_{ABSOLUTE\ STORAGE}$ | Device storage temperature when exceeded for any length of time. | -25 | 125 | °C |
| $T_{SUSTAINED\ STORAGE\ TIME\ AND\ TEMPERATURE}$ | The minimum/maximum device storage temperature for a sustained period of time. | -5 | 40 | °C |
| $T_{SHORT\ TERM\ STORAGE}$ | The ambient storage temperature and time for up to 72 hours. | -25 | 85 | °C |
| $RH_{SUSTAINED\ STORAGE}$ | The maximum device storage relative humidity for up to 30 months. | | 60% at 24 °C | |

***Notes:***
1. Specified temperatures are not to exceed values based on data collected. Exceptions for surface mount re-flow are specified by the applicable JEDEC standard. Non-adherence may affect processor reliability.
2. Component product device storage temperature qualification methods may follow JESD22-A119 (low temperature) and JESD22-A103 (high temperature) standards when applicable for volatile memory.
3. Component stress testing is conducted in conformance with JESD22-A104.
4. The JEDEC J-JSTD-020 moisture level rating and associated handling practices apply to all moisture sensitive devices removed from the moisture barrier bag.

#### 41.1.1.2 Post Board-Attach

The storage condition limits for the component once attached to the application board are not specified.

Intel does not conduct component-level certification assessments post board-attach given the multitude of attach methods and board types used by customers.

Provided as general guidance only, board-level Intel-branded products are specified and certified to meet the following temperature and humidity limits:

- Non-Operating Temperature Limit: -40 °C to 70 °C
- Humidity: 50% to 90%, non-condensing with a maximum wet-bulb of 28 °C

## 41.2 Normal Operating Conditions

### 41.2.1 Component Temperature

Table 41-2 specifies the normal operating temperature range for product SKUs that feature Commercial Temperature Range and for product SKUs that feature Extended Temperature Range. The list of product SKUs is in Chapter 1, "Introduction and Product Offerings."

All supply voltage requirements, input signal requirements, and output signal characteristics are specified for the normal operating temperature range of the device. The temperature range is specified in terms of the Package Junction Temperature (Tj) which is the temperature of the die active surface. For a platform board and chassis using the device, the design must maintain an operational Tj within the specified range.

**Table 41-2. Component Temperature Range**

| Parameter | Symbol | Minimum | Maximum | Unit | SoC Product SKU Feature |
|---|---|---|---|---|---|
| Package Junction Temperature | Tj[1] | 0 | 100 | °C | Commercial Temperature |
| | | -40[2] | 100 | °C | Extended Temperature |

1. All SKU thermal values are based on maintaining Tj minimum and maximum. Based on product SKU, the SoC has its $DTS_{MAX}$ value (also called REF_TEMP) set at the Intel factory so that the SoC does not throttle when running a TDP workload and meet the SoC $T_{CASE_MAX}$ specification. $DTS_{MAX}$ is described in Chapter 35, "Thermal Management".
2. While this temperature is lower than the specified minimum $T_{ABSOLUTE\ STORAGE}$ value, the Package Junction Temperature value assumes the part will be powered-on and hence won't stay at the minimum junction temperature for a long time

### 41.2.2 Supply Voltage and Current Requirements

Table 41-3 specifies the device voltage supply requirements for each of the voltage supply groups. These groups are described in Table 32-1, "Signal Descriptions" on page 893.

For the table below, the following definitions apply:

- **DC VR Error**
  This is the difference between the constant load VR output voltage and the measured output voltage averaged over time (tens of ms).

- **DC Tolerance**
  This is the three sigma of "DC VR Error", over the range of conditions, load and lifetime.

- **VR Ripple**
  This is the variation in the output voltage of a regulator due to the switching/sampling characteristics of the regulator. This ripple can be most easily measured by looking at the supply output under a constant load, across the range of loads. The ripple frequency will be at the bandwidth of the regulator.

- **AC Tolerance**
  The response to transients (Istep divided by di/dt) generated by the SOC or other platform components (e.g., feed-through from the VR input supply, which may have other loads).

- **Total Tolerance** = DC Tolerance + VR Ripple + AC Tolerance

- **AVP Sense Tolerance**
  This is the tolerance of the load line sensing circuitry over the range of conditions, load, and lifetime.

- **Total Tolerance With Load Line** = (IxRll) + DC Tolerance + VR Ripple + AC Tolerance + AVP Sense Tolerance = (IxRll) + Total Tolerance
  Where Rll is the Load Line Resistance.

The measurement points for the SoC VR specifications are at the SoC (the load). If access at the SoC is not practical, the measurements may be taken at the input of the VR sense signals, which is nearly identical to what the SoC receives.

## Table 41-3. Voltage Supply Requirements Under Normal Operating Conditions

| Group | Parameter | | Typical (V) | DC Tolerance | VR Ripple | Total Tolerance |
|---|---|---|---|---|---|---|
| VCC | Dynamic VID voltage for core circuitry (variable, switched internally) | SVID $V_{BOOT}$ | 1 | ± 6mV | N/A | Total Tolerance With Load Line ±50mV<br><br>See Table 41-4, "SoC VCC Load Line" |
| | | SVID | 0.52 - 1.24 | | | |
| | | Nominal | 1.15 | | | |
| VNN | Dynamic VID voltage for un-core circuitry (variable, always-on voltage supply) | SVID $V_{BOOT}$ | 1 | ± 0.5% | N/A | ± 6% |
| | | SVID | 0.65 - 1.24 | | | |
| | | Nominal | 1.05 | | | |
| VCCRAM | Dynamic VID for internal circuitry (variable, switched internally) | SVID $V_{BOOT}$ | 1 | ± 0.5% | N/A | ± 7% |
| | | SVID | 0.75 - 1.20 | | | |
| | | Nominal | 1.15 | | | |
| VDDQ | Static VID for internal Standard DDR4 circuitry (static, switched internally). *See Note below table.* | SVID $V_{BOOT}$ | 1 or 1.20 | ± 8mV | N/A | +60mV, -38mV<br><br>(1.260-1.162 V) |
| | | Nominal | 1.20 | | | |
| VCCREF | VCC reference voltage | | 1.24 | ± 2.1% | N/A | ± 50mV |
| V1P05 | Device internal circuitry (always-on voltage supply, switched internally) | | 1.05 | ± 0.6% | ± 3mV VR Ripple for frequencies less than 1MHz | ± 25mV |
| V1P80 | Device internal circuitry (always-on voltage supply, switched internally) | | 1.8 | ± 2.1% | N/A | ± 72mV |
| V3P30 | Device internal circuitry (always-on voltage supply, switched internally) | | 3.3 | ± 2% | N/A | ± 165mV |
| VRTC | All power states other than G3 Mechanical-Off state | | 3.3 | ± 2% | N/A | ± 165mV |
| VRTC | G3 State: From Real Time Clock (RTC) battery supply | | 2.0 - 3.0 | ± 2% | N/A | ± 165mV |

***Note:*** DDR4 DRAM Activating Power (VPP) supply (2.5 V nominal, 2.375 V min, 2.75 V max) goes to the DIMM component only and not to the SoC.

### Table 41-4. SoC VCC Load Line

| Vcc Tolerance / Load Line / Power | | Units |
|---|---|---|
| $VCC_{MAX}$ = | VID (V) – 2.3 mΩ * ICC (A) + 50 mV | V |
| $VCC_{TYP}$ = | VID (V) – 2.3 mΩ * ICC (A) | |
| $VCC_{MIN}$ = | VID (V) – 2.3 mΩ * ICC (A) - 50 mV | |
| PVCC = | ICC (A) * VCC (V) | W |
| Total Tolerance | ±50mV | V |

Adequate supply current is required for each device voltage group to ensure proper operation. The current requirements vary by product SKU group. The tables below show the characteristics of the current draw for all the SKUs in each group, over the supply voltage ranges shown in Table 41-3.

$ICC_{MAX}$ represents the worst-case transient current that the SoC draws from a given power supply. Platform board designers typically use this value to keep the board Voltage Regulator (VR) current below the VR Over-Current Protection (OCP) limit.

Thermal Design Current (TDC) represents the worst-case sustained current (DC equivalent) that the SoC draws from a given power supply. Platform board designers use this value for proper VR design including VR thermal design.

$DI/DT_{\mathbf{MAX}}$ represents the worst-case current change rate, with the worst-case current change step of $ICC_{\mathbf{STEP}}$.

These parameters determine the characteristics of each VR design, the copper layout connecting the VRs to the SoC; as well as determining the number, type, location, and size of capacitors for each VR.

*Note:* The $ICC_{TDC}$ values shown in the Supply Current Required tables below are representative numbers. $ICC_{TDC}$ values vary across parts and operating conditions. Only the $ICC_{MAX}$ values are guaranteed when the SoC product has achieved production release.

For the supported features of each SKU, please refer to Chapter 1, "Introduction and Product Offerings."

### Table 41-5. Supply Current Required - Server and Cloud Storage SKUs 0, 1, and 2 (See Table 1-1)

| Group | Parameter | $ICC_{TDC}$ | $ICC_{MAX}$ in S0 | $ICC_{MAX}$ in S5 | $ICC_{STEP}$ | $DI/DT_{MAX}$ [A/µs] |
|---|---|---|---|---|---|---|
| VCC | Dynamic VID voltage for core circuitry | 18A | 39A | 0A | 35A | 510 |
| VNN | Dynamic VID voltage for un-core circuitry | 4.2A | 4.5A | 2.3A | 3.5A | 50 |
| VCCRAM | Dynamic VID for internal circuitry | 6.5A | 8A | 0A | 3A | 10 |
| VDDQ | Static VID for internal Standard DDR4 circuitry | 2A | 5A | 0A | 3.5A | 18 |
| VCCREF | VCC reference voltage | 0.3A | 0.3A | 0A | 0.02 | 0.1 |
| V1P05 | Device internal circuitry | 10A | 12A | 2.3A | 5A | 20 |
| V1P80 | Device internal circuitry | 1A | 1A | 0.3A | 0.1A | 0.1 |
| V3P30 | Device internal circuitry | 0.5A | 0.5A | 0.4A | 0.5A | 1 |
| VRTC | All power states other than G3 Mechanical-Off state | 0.5mA | 0.5mA | | N/A | N/A |
| | G3 State: From Real Time Clock (RTC) battery supply | N/A | 9.5µA | | N/A | N/A |

### Table 41-6. Supply Current Required - Network & Enterprise Storage SKUs 3, 4, 5, 6, and 7 (See Table 1-2) and eTEMP SKUs 11 and 12 (See Table 1-3)

| Group | Parameter | $ICC_{TDC}$ | $ICC_{MAX}$ in S0 | $ICC_{MAX}$ in S5 | $ICC_{STEP}$ | $DI/DT_{MAX}$ [A/µs] |
|---|---|---|---|---|---|---|
| VCC | Dynamic VID voltage for core circuitry | 18A | 30A | 0A | 27A | 400 |
| VNN | Dynamic VID voltage for un-core circuitry | 6.7A | 7A | 2.3A | 5A | 80 |
| VCCRAM | Dynamic VID for internal circuitry | 6.5A | 8A | 0A | 3A | 10 |
| VDDQ | Static VID for internal Standard DDR4 circuitry | 2A | 5A | 0A | 3.5A | 18 |
| VCCREF | VCC reference voltage | 0.3A | 0.3A | 0A | 0.02A | 0.1 |
| V1P05 | Device internal circuitry | 10A | 12A | 2.3A | 5A | 20 |
| V1P80 | Device internal circuitry | 1A | 1A | 0.3A | 0.1A | 0.1 |
| V3P30 | Device internal circuitry | 0.5A | 0.5A | 0.4A | 0.5A | 1 |
| VRTC | All power states other than G3 Mechanical-Off state | 0.50mA | 0.50mA | | N/A | N/A |
| | G3 State: From Real Time Clock (RTC) battery supply | N/A | 9.5µA | | N/A | N/A |

### Table 41-7. Supply Current Required - Network & Enterprise Storage SKUs 8 and 9 (See Table 1-2)

| Group | Parameter | $ICC_{TDC}$ | $ICC_{MAX}$ in S0 | $ICC_{MAX}$ in S5 | $ICC_{STEP}$ | $DI/DT_{MAX}$ [A/µs] |
|---|---|---|---|---|---|---|
| VCC | Dynamic VID voltage for core circuitry | 7A | 11A | 0A | 10A | 120 |
| VNN | Dynamic VID voltage for un-core circuitry | 4.2A | 4.5A | 2A | 3.5A | 60 |
| VCCRAM | Dynamic VID for internal circuitry | 3A | 5A | 0A | 1.5A | 5 |
| VDDQ | Static VID for internal Standard DDR4 circuitry | 1A | 4A | 0A | 3.5A | 18 |
| VCCREF | VCC reference voltage | 0.3A | 0.3A | 0A | 0.02A | 0.1 |
| V1P05 | Device internal circuitry | 8A | 10A | 2.3A | 5A | 20 |
| V1P80 | Device internal circuitry | 1A | 1A | 0.3A | 0.1A | 0.1 |
| V3P30 | Device internal circuitry | 0.5A | 0.5A | 0.4A | 0.5A | 1 |
| VRTC | All power states other than G3 Mechanical-Off state | 0.50mA | 0.50mA | | N/A | N/A |
| | G3 State: From Real Time Clock (RTC) battery supply | N/A | 9.5µA | | N/A | N/A |

### Table 41-8. Supply Current Required - Network & Enterprise Storage SKU 10 (See Table 1-2) and eTEMP SKU 13 and 14 (See Table 1-3)

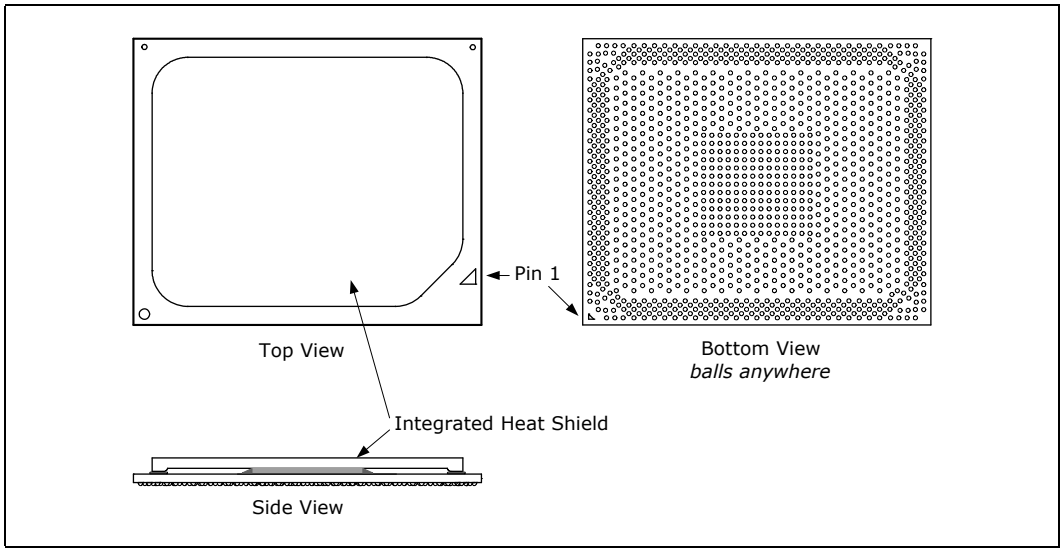
| Group | Parameter | $ICC_{TDC}$ | $ICC_{MAX}$ in S0 | $ICC_{MAX}$ in S5 | $ICC_{STEP}$ | $DI/DT_{MAX}$ [A/µs] |
|---|---|---|---|---|---|---|
| VCC | Dynamic VID voltage for core circuitry | 4A | 4.5A | 0A | 4A | 60 |
| VNN | Dynamic VID voltage for un-core circuitry | 4.2A | 4.5A | 2A | 3.5A | 60 |
| VCCRAM | Dynamic VID for internal circuitry | 2.5A | 3A | 0A | 1.5A | 5 |
| VDDQ | Static VID for internal Standard DDR4 circuitry | 1A | 4A | 0A | 3.5A | 18 |
| VCCREF | VCC reference voltage | 0.3A | 0.3A | 0A | 0.02A | 0.1 |
| V1P05 | Device internal circuitry | 8A | 10A | 2.3A | 5A | 20 |
| V1P80 | Device internal circuitry | 1A | 1A | 0.3A | 0.1A | 0.1 |
| V3P30 | Device internal circuitry | 0.5A | 0.5A | 0.4A | 0.5A | 1 |
| VRTC | All power states other than G3 Mechanical-Off state | 0.50mA | 0.50mA | | N/A | N/A |
| | G3 State: From Real Time Clock (RTC) battery supply | N/A | 9.5µA | | N/A | N/A |

§ §

# 42 Package Mechanical Overview

The SoC is manufactured as a 34 mm x 28 mm Flip-Chip Ball Grid Array (FBGA15) package and consists of a silicon die mounted face down on an organic substrate populated with 1310 solder balls (the term "pins" is also used in this document) on the bottom side. The balls have a mixed pitch—minimum pitch is 0.65 mm. The package contains an integrated heat spreader that is 30.36 mm x 24.36 mm, 1 mm thick. See Figure 42-1 for a representation of the SoC package.

**Figure 42-1. SoC Package Drawing**




§ §

# Volume 4: Registers

February 2018
Order Number: 337018-002

# 43 Introduction

This volume documents the registers accessible to software running in the IA Host Space. There is a register chapter for each PCI device/function in the Host Root Space. There are additional chapters for other SoC entities that have software-accessible registers. There is also a chapter for the SoC Model-Specific Registers (MSRs).

*Note:* Accessing unimplemented registers and/or writing reserved bits may result in undefined behavior on the platform.

## 43.1 Access to Registers in Host Root Space

The registers shown in this volume are those accessible in the SoC Host Root Configuration, Memory and I/O Spaces.

Details of the registers accessible in the Root Space of the Intel® Management Engine (Intel® ME) are not provided to SoC customers.

### 43.1.1 PCI Configuration Space

Access to the registers in PCI Configuration Space is performed through one of the two Configuration Access Methods (CAMs):

- I/O-indexed - PCI CAM (only the first 256 bytes of the function Configuration Space)
- Memory-mapped - PCI enhanced CAM (ECAM)

These two methods are explained in Section 3.3.1, "How to Access Registers in PCI Configuration Space" on page 185.

The descriptions for the Configuration Registers show the PCI Bus, Device, and Function numbers as Bbus#:Ddevice#:Ffunction#. An example is B0:D17:F0. All three of the numbers are decimal. The register offset address from the B:D:F: is also shown for each Configuration Register and is a hexadecimal value. The structure of the Configuration Registers are defined in the *PCI Local Bus Specification, Revision 3.0* and *PCI Express Base Specification Revision 3.0*.

*Note:* Some of the registers in Host Root Configuration Space can be accessed and altered through the SMBus-PECI interface by an external component such as a Baseboard Management Controller (BMC). See Chapter 30, "SMBus Controller - Platform Environment Control Interface".

#### 43.1.1.1 Unimplemented Devices/Functions and Registers

Read accesses in the Configuration Space is to unimplemented PCI functions and so the devices return all ones, emulating a Master Abort response. Note that there is no asynchronous error reporting that happens when a configuration-read transaction Master Aborts.

Write accesses in the Configuration Space is to unimplemented PCI functions and so the devices return a normal response.

Software must not attempt, or rely on, reads or writes to unimplemented registers or register bits. Unimplemented registers return all zeros when read. Writes to unimplemented registers are ignored by the SoC. For configuration writes to these registers (requires a Completion), the Completion is returned with a normal completion status (not Master-Aborted).

### 43.1.2 Memory Space—Relocatable Address

Access to this kind of register requires a base memory address defined in a Base Address Register (BAR). The register is accessed using an offset address that is added to the memory BAR. Memory-Mapped I/O (MMIO) registers are accessed in this fashion. A list of most of the Memory BARs is in Section 3.1.2, "Host Memory Space—Relocatable Memory-Mapped I/O" on page 169.

The register description for this type of register shows the name of the BAR and a hexadecimal offset value.

### 43.1.3 Memory Space—Fixed Address

Fixed-address registers in the Host Root Memory Space are accessed by software and the SoC hardware. The register locations are not relocatable and a BAR is not associated with the register access. The Host Root address map of these registers is in Section 3.1.1, "Host Memory Space—Regions and Boundaries" on page 166.

The register description for this type of register shows the memory address as a hexadecimal value.

### 43.1.4 I/O Space—Relocatable Address

Access to this kind of register requires a base I/O-Space address defined in a Base Address Register (BAR). The register is accessed using an offset address that is added to the I/O BAR. A list of the I/O BARs is in Section 3.2.2, "Host I/O Space—Relocatable I/O Addresses" on page 180.

The register description for this type of register shows the name of the I/O BAR and a hexadecimal offset value.

### 43.1.5 I/O Space—Fixed Address

Fixed-address registers in Host Root I/O Space (also known as I/O Ports) are accessed by the CPU when software executes the following instructions:

- Read: IN, INS INSB, INSW, and INSD
- Write: OUT, OUTS, OUTSB, OUTSW, and OUTSD

The descriptions of these I/O Port instructions are in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 2. The SoC Host Root address map of these registers is in Section 3.2.1, "Host I/O Space—Fixed I/O Addresses" on page 175

The register description for this type of register shows the I/O Port address as a hexadecimal value.

### 43.1.6 Sideband Registers

The SoC has an internal messaging mechanism that is used by the internal circuitry. This mechanism includes registers called Sideband Registers that are accessible to BIOS so it can configure and tune the system before the Operating System is loaded. BIOS accesses these Sideband Registers in the Host Root Memory Space. They are located as an offset address from the memory BAR defined as SBREG_BAR. A few Sideband Registers are accessible to BIOS as offsets from the Memory Controller Hub Base Address Register (MCHBAR). For more information, see Section 3.4, "Host Root Space - Host Sideband Register Access" on page 187.

The register description for this type of register shows the name of the memory BAR (either SBREG_BAR or MCHBAR) and a hexadecimal offset value.

### 43.1.7 Model-Specific Registers (MSR)

These registers are accessed by the CPU when software executes the following instructions:

- RDMSR - Read MSR
- WRMSR - Write MSR

The descriptions of the RDMSR and WRMSR instructions are in the *IA-32 Intel® Architecture Software Developer's Manual*, Volume 2. The access to the SoC Machine Check Registers is through MSRs. The SoC MSRs are all described in Chapter 69, "Model-Specific Registers (MSR)".

## 43.2 Register Description Terminology

### 43.2.1 Default Values

***Warning:*** BIOS must not rely on the register Default values provided in this document, and instead verify these values by first reading them.

Some Default values shown in the register-description tables vary by product SKU and are set to values at the factory that are different than published here. Generally, the Default values are set by hardware when the register or register fields are reset. During power-on situations, the original default values set by hardware may be written to different values before BIOS starts executing instructions. Things like Hard Pin Straps, customer-selected Soft Straps, the SoC power-management controller, the Intel® Management Engine (Intel® ME) and the Innovation Engine (IE) may alter the default values before BIOS starts executing instructions. By the time that the Operating System is loaded, BIOS could have also modified what was the Default value.

## 43.2.2 Software Access Attributes

Table 43-1 is a comprehensive list of software-access attributes used in the register descriptions.

### Table 43-1. Register Attributes Definitions

| Attribute | Description |
|---|---|
| RO | **Read Only:** These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only which includes pin strapping, soft straps, and internal microcode. |
| RW | **Read / Write:** These bits can be read and written by software. |
| RO/C<br>RO/C/V | **Read Clear Variant:** These bits can be read by software, and the act of reading them automatically clears them. Hardware is responsible for writing these bits, and therefore the / V modifier is implied. |
| WO/1S | **Write 1 to Set:** Writing a '1' to these bits will set them to '1.' Writing '0' will have no effect. Reading will return indeterminate values. |
| WO/1C | **Write 1 to Clear:** Writing a '1' to these bits will set them to '0.' Writing '1' will have no effect. Reading will return indeterminate values. |
| RW/O | **Read / Write Once:** These bits can be read by software. After reset, these bits can only be written by software once, after which the bits becomes "Read Only." |
| RW/L | **Read / Write Lock:** These bits can be read and written by software. The bits can be made to be "Read Only" via a separate configuration bit or other logic. |
| RW/1C | **Read / Write 1 to Clear:** These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. |
| RW/0C | **Read / Write 0 to Clear:** These bits can be read and cleared by software. Writing a '0' to a bit clears it while writing a '1' has no effect. |
| RO/P | **RO Sticky:** These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only. These bits are only re-initialized to their default value by a PWRGOOD reset. |
| RW/1S | **Read, Write 1 to Set:** These bits can be read. Writing a '1' to a given bit will set it to '1.' Writing a 0 to a given bit will have no effect. It is not possible for software to set a bit to '0.' The '1'-to-'0' transition can only be performed by hardware. These registers are implicitly /V modified. |
| RW/P | **R / W Sticky (Persistent):** These bits can be read and written by software. These bits are only re- initialized to their default value by a PWRGOOD reset. |
| RW/1C/P | **R / W1C Sticky:** These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. These bits are only re-initialized to their default value by a PWRGOOD reset. |
| RW/O/P | If a register is both sticky (persistent) and "once," then the sticky value applies to both the register value and the "once" characteristic. Only a PWRGOOD reset will reset both the value and the "once" so that the register can be written to again. |
| RW/V<br>RO/V | These bits may be modified by hardware. Software cannot expect the values to stay unchanged. |
| RW/V/P<br>RO/V/P | These bits may be modified by hardware and are Sticky. Software cannot expect the values to stay unchanged. These bits are re-initialized to their default values by a PWRGOOD reset. |
| RW/P/L | If a register is both sticky and locked, then the sticky behavior only applies to the value. The sticky behavior of the lock is determined by the register that controls the lock. |
| RW/V/P/L | These bits can be read or written by software and may be modified by hardware. Software cannot expect the values to stay unchanged. These bits are re-initialized to their default values by a PWRGOOD reset. If a register is both sticky and locked, then the sticky behavior only applies to the value. The sticky behavior of the lock is determined by the register that controls the lock. |
| RSV | **Reserved:** These bits are reserved for future expansion and their value must not be modified by software. When writing these bits, software must preserve the value read. |
| NA | **No Access:** These bits are not accessible from the given source. Writes are dropped and Reads return '0.' |

### 43.2.3 Hexadecimal, Binary and Decimal Numbers

Base 16 numbers are represented by a string of hexadecimal digits followed by the character “h” (for example, F82Eh). A hexadecimal digit is a character from the following set: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Hexadecimal numbers can also be shown using an “x” character (for example 0x2A).

Base 2 (binary) numbers are represented by a string of 1s and 0s, sometimes followed by the character “b” (for example, 1010001b). The “b” designation is only used in situations where confusion as to the type of the number might arise.

Base 10 numbers are represented by a string of decimal digits followed by the character “d” (for example, 23d). The “d” designation is only used in situations where confusion as to the type of the number might arise.

§ §

# 44 System Agent - B0, D0, F0

## 44.1 Introduction and Index

The host-accessible registers for the System Agent are described here. The legend for the register access codes are described in Chapter 43, "Introduction."

### 44.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 0 (decimal), Function 0. The offset addresses are listed.

**Table 44-1. Summary of PCI Configuration Registers—0/0/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19908086 | "Device ID and Vendor ID Register (DEVICE_ID_VENDOR_ID_0_0_0_PCI)—Offset 0h" on page 1103 |
| 4 | 4 | 00000007 | "PCI Status and PCI Command Register (PCI_STATUS_COMMAND_0_0_0_PCI)—Offset 4h" on page 1103 |
| 8 | 4 | 06000000 | "PCI Revision ID and PCI Class Code Register (REVISION_ID_CLASS_CODE_0_0_0_PCI)—Offset 8h" on page 1104 |
| C | 4 | 00800000 | "Master Latency Timer and Header Type Register (MASTER_LATENCY_TIME_0_0_0_PCI)—Offset Ch" on page 1104 |
| 2C | 4 | 00000000 | "PCI Subsystem Vendor ID and PCI Subsystem ID (SVID_SID_0_0_0_PCI)—Offset 2Ch" on page 1105 |
| 34 | 4 | 000000E0 | "Capability Register Pointer (CAPPTR_0_0_0_PCI)—Offset 34h" on page 1105 |
| 48 | 4 | 00000000 | "Memory Controller Hub Base Address Register (MCHBAR_LO_0_0_0_PCI)—Offset 48h" on page 1106 |
| 4C | 4 | 00000000 | "Memory Controller Hub Base Address Register (MCHBAR_HI_0_0_0_PCI)—Offset 4Ch" on page 1106 |
| 54 | 4 | 00000033 | "Device Enable Register (DEVEN_0_0_0_PCI)—Offset 54h" on page 1107 |
| 60 | 4 | 00000000 | "PCI Express Enhanced Configuration Range Base Address Low (PCIEXBAR_LO_0_0_0_PCI)—Offset 60h" on page 1108 |
| 64 | 4 | 00000000 | "PCI Express Enhanced Configuration Range Base Address High (PCIEXBAR_HI_0_0_0_PCI)—Offset 64h" on page 1109 |
| A8 | 4 | 00000000 | "Top of Upper Usable DRAM Low (TOUUD_LO_0_0_0_PCI)—Offset A8h" on page 1110 |
| AC | 4 | 00000000 | "Top of Upper Usable DRAM High (TOUUD_HI_0_0_0_PCI)—Offset ACh" on page 1111 |
| B0 | 4 | 00000000 | "Base of Data Stolen Memory (BDSM_0_0_0_PCI)—Offset B0h" on page 1112 |
| B4 | 4 | 00000000 | "Base of Graphics Stolen Memory (BGSM_0_0_0_PCI)—Offset B4h" on page 1113 |
| B8 | 4 | 00100000 | "Top Segment Memory Base (TSEGMB_0_0_0_PCI)—Offset B8h" on page 1114 |
| BC | 4 | 00100000 | "Top of Lower Usable DRAM (TOLUD_0_0_0_PCI)—Offset BCh" on page 1115 |
| DC | 4 | 00000000 | "Scratchpad (SKPD_0_0_0_PCI)—Offset DCh" on page 1115 |
| E0 | 4 | 010C0009 | "Capability ID0 Capability Control (CAPID0_CAPCTRL0_0_0_0_PCI)—Offset E0h" on page 1116 |
| E4 | 4 | 00000000 | "Capability ID0 A (CAPID0_A_0_0_0_PCI)—Offset E4h" on page 1117 |
| E8 | 4 | 00000000 | "Capability ID0 B (CAPID0_B_0_0_0_PCI)—Offset E8h" on page 1118 |
| F4 | 4 | 00000000 | "Manufacturer ID Register (MANUFACTURER_ID_0_0_0_PCI)—Offset F4h" on page 1119 |

### 44.1.2 Host Memory Space—MCHBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 44-2. Summary of Memory Mapped I/O Registers for C-Unit—MCHBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| **C-Unit Registers** | | | |
| 6C80 | 8 | 0000000000000000 | "Default VTd Base Address Register (DEFVTDBAR_0_0_0_MCHBAR_C)—Offset 6C80h" on page 1120 |

## Table 44-3. Summary of Memory Mapped I/O Registers for A-Unit—MCHBAR

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 647C | 4 | 00000000 | "Spare BIOS (A_CR_SPARE_BIOS_MCHBAR)—Offset 647Ch" on page 1121 |
| 6588 | 4 | 00000000 | "Uncorrectable Error Status Register (A_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6588h" on page 1122 |
| 658C | 4 | FFFFFFFF | "Uncorrectable Error Mask Register (A_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 658Ch" on page 1125 |
| 6590 | 4 | 00000000 | "Uncorrectable Error Severity Register (A_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6590h" on page 1127 |
| 6594 | 4 | 00000000 | "First Uncorrectable Error Register (A_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6594h" on page 1129 |
| 6598 | 4 | 00000000 | "Next Uncorrectable Error Register (A_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6598h" on page 1132 |
| 659C | 4 | 00000000 | "Uncorrectable Error Select Register (A_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 659Ch" on page 1135 |
| 65A0 | 4 | 00000000 | "Uncorrectable Error Count (A_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 65A0h" on page 1137 |
| 65A4 | 4 | 00000000 | "Correctable Error Status Register (A_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 65A4h" on page 1138 |
| 65A8 | 4 | FFFFFFFF | "Correctable Error Mask Register (A_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 65A8h" on page 1141 |
| 65AC | 4 | 00000000 | "First Correctable Error Register (A_CR_CORFERRSTS_0_0_0_MCHBAR)—Offset 65ACh" on page 1143 |
| 65B0 | 4 | 00000000 | "Next Correctable Error Register (A_CR_CORNERRSTS_0_0_0_MCHBAR)—Offset 65B0h" on page 1146 |
| 65B4 | 4 | FFFFFFFF | "Correctable Error Count Select (A_CR_CORERRCNTSEL_0_0_0_MCHBAR)—Offset 65B4h" on page 1149 |
| 65B8 | 4 | 00000000 | "Correctable Error Count (A_CR_CORERRCNT_0_0_0_MCHBAR)—Offset 65B8h" on page 1151 |
| 65C0 | 8 | 0000000000000000 | "Slice and Channel Hash (A_CR_SLICE_CHANNEL_HASH_0_0_0_MCHBAR)—Offset 65C0h" on page 1152 |
| 65D0 | 4 | 00000000 | "A_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR—Offset 65D0h" on page 1154 |
| 65D4 | 4 | 00000000 | "A_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR—Offset 65D4h" on page 1154 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 1 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6800 | 4 | 00000007 | "B-Unit Miscellaneous Configuration (B_CR_BMISC_0_0_0_MCHBAR)—Offset 6800h" on page 1155 |
| 6868 | 4 | 00000000 | "Slice 0 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE0)—Offset 6868h" on page 1155 |
| 686C | 4 | 00000000 | "Slice 1 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE1)—Offset 686Ch" on page 1155 |
| 6870 | 4 | 00000000 | "IMR0 Base (B_CR_BIMR0BASE_0_0_0_MCHBAR)—Offset 6870h" on page 1156 |
| 6874 | 4 | 00000000 | "IMR0 Mask (B_CR_BIMR0MASK_0_0_0_MCHBAR)—Offset 6874h" on page 1157 |
| 6878 | 8 | 00000C0061010202 | "IMR0 Control Policy (B_CR_BIMR0CP_0_0_0_MCHBAR)—Offset 6878h" on page 1157 |
| 6880 | 8 | 0000000000000000 | "IMR0 Read Access Policy (B_CR_BIMR0RAC_0_0_0_MCHBAR)—Offset 6880h" on page 1158 |
| 6888 | 8 | 0000000000000000 | "IMR0 Write Access Policy (B_CR_BIMR0WAC_0_0_0_MCHBAR)—Offset 6888h" on page 1161 |
| 6890 | 4 | 00000000 | "B_CR_BIMR1BASE_0_0_0_MCHBAR—Offset 6890h" on page 1164 |
| 6894 | 4 | 00000000 | "IMR1 Mask (B_CR_BIMR1MASK_0_0_0_MCHBAR)—Offset 6894h" on page 1165 |
| 6898 | 8 | 00000C0061010202 | "IMR1 Control Policy (B_CR_BIMR1CP_0_0_0_MCHBAR)—Offset 6898h" on page 1165 |
| 68A0 | 8 | 0000000000000000 | "IMR1 Read Access Policy (B_CR_BIMR1RAC_0_0_0_MCHBAR)—Offset 68A0h" on page 1166 |
| 68A8 | 8 | 0000000000000000 | "IMR1 Write Access Policy (B_CR_BIMR1WAC_0_0_0_MCHBAR)—Offset 68A8h" on page 1169 |
| 68B0 | 4 | 00000000 | "Base 0 IMR2 Base (B_CR_BIMR2BASE_0_0_0_MCHBAR)—Offset 68B0h" on page 1172 |
| 68B4 | 4 | 00000000 | "IMR2 Mask (B_CR_BIMR2MASK_0_0_0_MCHBAR)—Offset 68B4h" on page 1173 |
| 68B8 | 8 | 00000C0061010202 | "IMR2 Control Policy (B_CR_BIMR2CP_0_0_0_MCHBAR)—Offset 68B8h" on page 1173 |
| 68C0 | 8 | 0000000000000000 | "IMR2 Read Access Policy (B_CR_BIMR2RAC_0_0_0_MCHBAR)—Offset 68C0h" on page 1174 |
| 68C8 | 8 | 0000000000000000 | "IMR2 Write Access Policy (B_CR_BIMR2WAC_0_0_0_MCHBAR)—Offset 68C8h" on page 1177 |
| 68D0 | 4 | 00000000 | "IMR3 Base (B_CR_BIMR3BASE_0_0_0_MCHBAR)—Offset 68D0h" on page 1180 |
| 68D4 | 4 | 00000000 | "IMR3 Mask (B_CR_BIMR3MASK_0_0_0_MCHBAR)—Offset 68D4h" on page 1181 |
| 68D8 | 8 | 00000C0061010202 | "IMR3 Control Policy (B_CR_BIMR3CP_0_0_0_MCHBAR)—Offset 68D8h" on page 1181 |
| 68E0 | 8 | 0000000000000000 | "IMR3 Read Access Policy (B_CR_BIMR3RAC_0_0_0_MCHBAR)—Offset 68E0h" on page 1182 |
| 68E8 | 8 | 0000000000000000 | "IMR3 Write Access Policy (B_CR_BIMR3WAC_0_0_0_MCHBAR)—Offset 68E8h" on page 1185 |
| 68F0 | 4 | 00000000 | "IMR4 Base (B_CR_BIMR4BASE_0_0_0_MCHBAR)—Offset 68F0h" on page 1188 |
| 68F4 | 4 | 00000000 | "IMR4 Mask (B_CR_BIMR4MASK_0_0_0_MCHBAR)—Offset 68F4h" on page 1189 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 2 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 68F8 | 8 | 00000C0061010202 | "B-Unit IMR4 Control Policy (B_CR_BIMR4CP_0_0_0_MCHBAR)—Offset 68F8h" on page 1189 |
| 6900 | 8 | 0000000000000000 | "IMR4 Read Access Policy (B_CR_BIMR4RAC_0_0_0_MCHBAR)—Offset 6900h" on page 1190 |
| 6908 | 8 | 0000000000000000 | "IMR4 Write Access Policy (B_CR_BIMR4WAC_0_0_0_MCHBAR)—Offset 6908h" on page 1193 |
| 6910 | 4 | 00000000 | "IMR5 Base (B_CR_BIMR5BASE_0_0_0_MCHBAR)—Offset 6910h" on page 1196 |
| 6914 | 4 | 00000000 | "IMR5 Mask (B_CR_BIMR5MASK_0_0_0_MCHBAR)—Offset 6914h" on page 1197 |
| 6918 | 8 | 00000C0061010202 | "IMR5 Control Policy (B_CR_BIMR5CP_0_0_0_MCHBAR)—Offset 6918h" on page 1197 |
| 6920 | 8 | 0000000000000000 | "IMR5 Read Access Policy (B_CR_BIMR5RAC_0_0_0_MCHBAR)—Offset 6920h" on page 1198 |
| 6928 | 8 | 0000000000000000 | "IMR5 Write Access Policy (B_CR_BIMR5WAC_0_0_0_MCHBAR)—Offset 6928h" on page 1201 |
| 6930 | 4 | 00000000 | "IMR6 Base (B_CR_BIMR6BASE_0_0_0_MCHBAR)—Offset 6930h" on page 1204 |
| 6934 | 4 | 00000000 | "IMR6 Mask (B_CR_BIMR6MASK_0_0_0_MCHBAR)—Offset 6934h" on page 1205 |
| 6938 | 8 | 00000C0061010202 | "IMR6 Control Policy (B_CR_BIMR6CP_0_0_0_MCHBAR)—Offset 6938h" on page 1205 |
| 6940 | 8 | 0000000000000000 | "IMR6 Read Access Policy (B_CR_BIMR6RAC_0_0_0_MCHBAR)—Offset 6940h" on page 1206 |
| 6948 | 8 | 0000000000000000 | "IMR6 Write Access Policy (B_CR_BIMR6WAC_0_0_0_MCHBAR)—Offset 6948h" on page 1209 |
| 6950 | 4 | 00000000 | "IMR7 Base (B_CR_BIMR7BASE_0_0_0_MCHBAR)—Offset 6950h" on page 1212 |
| 6954 | 4 | 00000000 | "IMR7 Mask (B_CR_BIMR7MASK_0_0_0_MCHBAR)—Offset 6954h" on page 1213 |
| 6958 | 8 | 00000C0061010202 | "IMR7 Control Policy (B_CR_BIMR7CP_0_0_0_MCHBAR)—Offset 6958h" on page 1213 |
| 6960 | 8 | 0000000000000000 | "IMR7 Read Access Policy (B_CR_BIMR7RAC_0_0_0_MCHBAR)—Offset 6960h" on page 1214 |
| 6968 | 8 | 0000000000000000 | "IMR7 Write Access Policy (B_CR_BIMR7WAC_0_0_0_MCHBAR)—Offset 6968h" on page 1217 |
| 6970 | 4 | 00000000 | "IMR8 Base (B_CR_BIMR8BASE_0_0_0_MCHBAR)—Offset 6970h" on page 1220 |
| 6974 | 4 | 00000000 | "IMR8 Mask (B_CR_BIMR8MASK_0_0_0_MCHBAR)—Offset 6974h" on page 1221 |
| 6978 | 8 | 00000C0061010202 | "IMR8 Control Policy (B_CR_BIMR8CP_0_0_0_MCHBAR)—Offset 6978h" on page 1221 |
| 6980 | 8 | 0000000000000000 | "IMR8 Read Access Policy (B_CR_BIMR8RAC_0_0_0_MCHBAR)—Offset 6980h" on page 1222 |
| 6988 | 8 | 0000000000000000 | "IMR8 Write Access Policy (B_CR_BIMR8WAC_0_0_0_MCHBAR)—Offset 6988h" on page 1225 |
| 6990 | 4 | 00000000 | "IMR9 Base (B_CR_BIMR9BASE_0_0_0_MCHBAR)—Offset 6990h" on page 1228 |
| 6994 | 4 | 00000000 | "IMR9 Mask (B_CR_BIMR9MASK_0_0_0_MCHBAR)—Offset 6994h" on page 1229 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 3 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6998 | 8 | 00000C0061010202 | "IMR9 Control Policy (B_CR_BIMR9CP_0_0_0_MCHBAR)—Offset 6998h" on page 1229 |
| 69A0 | 8 | 0000000000000000 | "IMR9 Read Access Policy (B_CR_BIMR9RAC_0_0_0_MCHBAR)—Offset 69A0h" on page 1230 |
| 69A8 | 8 | 0000000000000000 | "IMR9 Write Access Policy (B_CR_BIMR9WAC_0_0_0_MCHBAR)—Offset 69A8h" on page 1233 |
| 69B0 | 4 | 00000000 | "IMR10 Base (B_CR_BIMR10BASE_0_0_0_MCHBAR)—Offset 69B0h" on page 1236 |
| 69B4 | 4 | 00000000 | "B_CR_BIMR10MASK_0_0_0_MCHBAR—Offset 69B4h" on page 1237 |
| 69B8 | 8 | 00000C0061010202 | "IMR10 Control Policy (B_CR_BIMR10CP_0_0_0_MCHBAR)—Offset 69B8h" on page 1237 |
| 69C0 | 8 | 0000000000000000 | "IMR10 Read Access Policy (B_CR_BIMR10RAC_0_0_0_MCHBAR)—Offset 69C0h" on page 1238 |
| 69C8 | 8 | 0000000000000000 | "IMR10 Write Access Policy (B_CR_BIMR10WAC_0_0_0_MCHBAR)—Offset 69C8h" on page 1241 |
| 69D0 | 4 | 00000000 | "IMR11 Base (B_CR_BIMR11BASE_0_0_0_MCHBAR)—Offset 69D0h" on page 1244 |
| 69D4 | 4 | 00000000 | "IMR11 Mask (B_CR_BIMR11MASK_0_0_0_MCHBAR)—Offset 69D4h" on page 1245 |
| 69D8 | 8 | 00000C0061010202 | "IMR11 Control Policy (B_CR_BIMR11CP_0_0_0_MCHBAR)—Offset 69D8h" on page 1245 |
| 69E0 | 8 | 0000000000000000 | "IMR11 Read Access Policy (B_CR_BIMR11RAC_0_0_0_MCHBAR)—Offset 69E0h" on page 1246 |
| 69E8 | 8 | 0000000000000000 | "IMR11 Write Access Policy (B_CR_BIMR11WAC_0_0_0_MCHBAR)—Offset 69E8h" on page 1249 |
| 69F0 | 4 | 00000000 | "IMR12 Base (B_CR_BIMR12BASE_0_0_0_MCHBAR)—Offset 69F0h" on page 1252 |
| 69F4 | 4 | 00000000 | "IMR12 Mask (B_CR_BIMR12MASK_0_0_0_MCHBAR)—Offset 69F4h" on page 1253 |
| 69F8 | 8 | 00000C0061010202 | "IMR12 Control Policy (B_CR_BIMR12CP_0_0_0_MCHBAR)—Offset 69F8h" on page 1253 |
| 6A00 | 8 | 0000000000000000 | "IMR12 Read Access Policy (B_CR_BIMR12RAC_0_0_0_MCHBAR)—Offset 6A00h" on page 1254 |
| 6A08 | 8 | 0000000000000000 | "IMR12 Write Access Policy (B_CR_BIMR12WAC_0_0_0_MCHBAR)—Offset 6A08h" on page 1257 |
| 6A10 | 4 | 00000000 | "IMR13 Base (B_CR_BIMR13BASE_0_0_0_MCHBAR)—Offset 6A10h" on page 1260 |
| 6A14 | 4 | 00000000 | "IMR13 Mask (B_CR_BIMR13MASK_0_0_0_MCHBAR)—Offset 6A14h" on page 1261 |
| 6A18 | 8 | 00000C0061010202 | "IMR13 Control Policy (B_CR_BIMR13CP_0_0_0_MCHBAR)—Offset 6A18h" on page 1261 |
| 6A20 | 8 | 0000000000000000 | "IMR13 Read Access Policy (B_CR_BIMR13RAC_0_0_0_MCHBAR)—Offset 6A20h" on page 1262 |
| 6A28 | 8 | 0000000000000000 | "IMR13 Write Access Policy (B_CR_BIMR13WAC_0_0_0_MCHBAR)—Offset 6A28h" on page 1265 |
| 6A30 | 4 | 00000000 | "IMR14 Base (B_CR_BIMR14BASE_0_0_0_MCHBAR)—Offset 6A30h" on page 1268 |
| 6A34 | 4 | 00000000 | "IMR14 Mask (B_CR_BIMR14MASK_0_0_0_MCHBAR)—Offset 6A34h" on page 1269 |
| 6A38 | 8 | 00000C0061010202 | "IMR14 Control Policy (B_CR_BIMR14CP_0_0_0_MCHBAR)—Offset 6A38h" on page 1269 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 4 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6A40 | 8 | 0000000000000000 | "IMR14 Read Access Policy (B_CR_BIMR14RAC_0_0_0_MCHBAR)—Offset 6A40h" on page 1270 |
| 6A48 | 8 | 0000000000000000 | "IMR14 Write Access Policy (B_CR_BIMR14WAC_0_0_0_MCHBAR)—Offset 6A48h" on page 1273 |
| 6A50 | 4 | 00000000 | "IMR15 Base (B_CR_BIMR15BASE_0_0_0_MCHBAR)—Offset 6A50h" on page 1276 |
| 6A54 | 4 | 00000000 | "IMR15 Mask (B_CR_BIMR15MASK_0_0_0_MCHBAR)—Offset 6A54h" on page 1277 |
| 6A58 | 8 | 00000C0061010202 | "IMR15 Control Policy (B_CR_BIMR15CP_0_0_0_MCHBAR)—Offset 6A58h" on page 1277 |
| 6A60 | 8 | 0000000000000000 | "IMR15 Read Access Policy (B_CR_BIMR15RAC_0_0_0_MCHBAR)—Offset 6A60h" on page 1278 |
| 6A68 | 8 | 0000000000000000 | "IMR15 Write Access Policy (B_CR_BIMR15WAC_0_0_0_MCHBAR)—Offset 6A68h" on page 1281 |
| 6A70 | 4 | 00000000 | "IMR16 Base (B_CR_BIMR16BASE_0_0_0_MCHBAR)—Offset 6A70h" on page 1284 |
| 6A74 | 4 | 00000000 | "IMR16 Mask (B_CR_BIMR16MASK_0_0_0_MCHBAR)—Offset 6A74h" on page 1285 |
| 6A78 | 8 | 00000C0061010202 | "IMR16 Control Policy (B_CR_BIMR16CP_0_0_0_MCHBAR)—Offset 6A78h" on page 1285 |
| 6A80 | 8 | 0000000000000000 | "IMR16 Read Access Policy (B_CR_BIMR16RAC_0_0_0_MCHBAR)—Offset 6A80h" on page 1286 |
| 6A88 | 8 | 0000000000000000 | "IMR16 Write Access Policy (B_CR_BIMR16WAC_0_0_0_MCHBAR)—Offset 6A88h" on page 1289 |
| 6A90 | 4 | 00000000 | "IMR17 Base (B_CR_BIMR17BASE_0_0_0_MCHBAR)—Offset 6A90h" on page 1292 |
| 6A94 | 4 | 00000000 | "IMR17 Mask (B_CR_BIMR17MASK_0_0_0_MCHBAR)—Offset 6A94h" on page 1293 |
| 6A98 | 8 | 00000C0061010202 | "IMR17 Control Policy (B_CR_BIMR17CP_0_0_0_MCHBAR)—Offset 6A98h" on page 1293 |
| 6AA0 | 8 | 0000000000000000 | "IMR17 Read Access Policy (B_CR_BIMR17RAC_0_0_0_MCHBAR)—Offset 6AA0h" on page 1294 |
| 6AA8 | 8 | 0000000000000000 | "IMR17 Write Access Policy (B_CR_BIMR17WAC_0_0_0_MCHBAR)—Offset 6AA8h" on page 1297 |
| 6AB0 | 4 | 00000000 | "IMR18 Base (B_CR_BIMR18BASE_0_0_0_MCHBAR)—Offset 6AB0h" on page 1300 |
| 6AB4 | 4 | 00000000 | "IMR18 Mask (B_CR_BIMR18MASK_0_0_0_MCHBAR)—Offset 6AB4h" on page 1301 |
| 6AB8 | 8 | 00000C0061010202 | "IMR18 Control Policy (B_CR_BIMR18CP_0_0_0_MCHBAR)—Offset 6AB8h" on page 1301 |
| 6AC0 | 8 | 0000000000000000 | "IMR18 Read Access Policy (B_CR_BIMR18RAC_0_0_0_MCHBAR)—Offset 6AC0h" on page 1302 |
| 6AC8 | 8 | 0000000000000000 | "IMR18 Write Access Policy (B_CR_BIMR18WAC_0_0_0_MCHBAR)—Offset 6AC8h" on page 1305 |
| 6AD0 | 4 | 00000000 | "IMR19 Base (B_CR_BIMR19BASE_0_0_0_MCHBAR)—Offset 6AD0h" on page 1308 |
| 6AD4 | 4 | 00000000 | "IMR19 Mask (B_CR_BIMR19MASK_0_0_0_MCHBAR)—Offset 6AD4h" on page 1309 |
| 6AD8 | 8 | 00000C0061010202 | "IMR19 Control Policy (B_CR_BIMR19CP_0_0_0_MCHBAR)—Offset 6AD8h" on page 1309 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 5 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6AE0 | 8 | 0000000000000000 | "IMR19 Read Access Policy (B_CR_BIMR19RAC_0_0_0_MCHBAR)—Offset 6AE0h" on page 1310 |
| 6AE8 | 8 | 0000000000000000 | "IMR19 Write Access Policy (B_CR_BIMR19WAC_0_0_0_MCHBAR)—Offset 6AE8h" on page 1313 |
| 6C24 | 4 | 00000000 | "TPM Selector (B_CR_TPM_SELECTOR_0_0_0_MCHBAR)—Offset 6C24h" on page 1316 |
| 6C80 | 8 | 0000000000000000 | "Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR)—Offset 6C80h" on page 1316 |
| 6D4C | 4 | 04040404 | "B-Unit Arbiter Control BARBCTRL0 (B_CR_BARBCTRL0)—Offset 6D4Ch" on page 1317 |
| 6D50 | 4 | 04040404 | "B-Unit Arbiter Control BARBCTRL1 (B_CR_BARBCTRL1)—Offset 6D50h" on page 1318 |
| 6D54 | 4 | 04040404 | "B-Unit Scheduler Control (B_CR_BSCHWT0)—Offset 6D54h" on page 1319 |
| 6D58 | 4 | 04040404 | "B-Unit Scheduler Control (B_CR_BSCHWT1)—Offset 6D58h" on page 1320 |
| 6D5C | 4 | 04040404 | "B-Unit Scheduler Control (B_CR_BSCHWT2)—Offset 6D5Ch" on page 1321 |
| 6D60 | 4 | 04040404 | "B-Unit Scheduler Control (B_CR_BSCHWT3)—Offset 6D60h" on page 1322 |
| 6D64 | 4 | FF010000 | "B-Unit Flush Control (B_CR_BWFLUSH)—Offset 6D64h" on page 1323 |
| 6D68 | 4 | 00000404 | "B-Unit Flush Weights (B_CR_BFLWT)—Offset 6D68h" on page 1323 |
| 6D6C | 4 | 80003F0F | "Weighted Scheduling Control of High Priority ISOC and Other Requests (B_CR_BISOCWT)—Offset 6D6Ch" on page 1324 |
| 6D70 | 4 | 000F0014 | "B-Unit Control (B_CR_BCTRL2)—Offset 6D70h" on page 1325 |
| 6E04 | 4 | 00000000 | "Uncorrectable Error Status Register (B_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6E04h" on page 1326 |
| 6E08 | 4 | FFFFFFFF | "Uncorrectable Error Mask Register (B_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 6E08h" on page 1329 |
| 6E0C | 4 | 00000000 | "Uncorrectable Error Severity Register (B_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6E0Ch" on page 1331 |
| 6E10 | 4 | 00000000 | "First Uncorrectable Error Register (B_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6E10h" on page 1333 |
| 6E14 | 4 | 00000000 | "Next Uncorrectable Error Register (B_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6E14h" on page 1336 |
| 6E18 | 4 | 00000000 | "Uncorrectable Error Select Register (B_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 6E18h" on page 1339 |
| 6E1C | 4 | 00000000 | "Uncorrectable Error Count (B_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 6E1Ch" on page 1341 |
| 6E20 | 4 | 00000000 | "Correctable Error Status Register (B_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 6E20h" on page 1342 |
| 6E24 | 4 | FFFFFFFF | "Correctable Error Mask Register (B_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 6E24h" on page 1345 |
| 6E28 | 4 | 00000000 | "Data Parity Header Log Low (B_CR_DPHDRLOG_LO_0_0_0_MCHBAR)—Offset 6E28h" on page 1347 |
| 6E2C | 4 | 00000000 | "Data Parity Header Log High (B_CR_DPHDRLOG_HI_0_0_0_MCHBAR)—Offset 6E2Ch" on page 1347 |
| 6E30 | 4 | 00000000 | "Data Parity Address Log Low (B_CR_DPADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E30h" on page 1347 |
| 6E34 | 4 | 00000000 | "Data Parity Address Log High (B_CR_DPADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E34h" on page 1348 |
| 6E38 | 4 | 00000000 | "D-Unit Error Address Log (B_CR_DERRADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E38h" on page 1348 |

## Table 44-4. Summary of Memory Mapped I/O Registers for B-Unit—MCHBAR (Sheet 6 of 6)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6E3C | 4 | 00000000 | "D-Unit Error Address Log (B_CR_DERRADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E3Ch" on page 1348 |
| 6E40 | 4 | 00000000 | "Asymmetric Memory Region 0 (B_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR)—Offset 6E40h" on page 1349 |
| 6E44 | 4 | 00000000 | "Asymmetric Memory Region 1 (B_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR)—Offset 6E44h" on page 1349 |
| 6E48 | 4 | 00000001 | "B-Unit Machine Check Mode Low (B_CR_BMCMODE_LOW)—Offset 6E48h" on page 1350 |
| 6E4C | 4 | 00000000 | "B-Unit Machine Check Mode High (B_CR_BMCMODE_HIGH)—Offset 6E4Ch" on page 1350 |

## Table 44-5. Summary of Memory Mapped I/O Registers for P-Unit—MCHBAR (Sheet 1 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 7014 | 4 | 00000000 | "PWR_LIMIT_MISC_0_0_0_MCHBAR (P_CR_PWR_LIMIT_MISC_0_0_0_MCHBAR)—Offset 7014h" on page 1351 |
| 7018 | 4 | 00000000 | "CAPABILITY_ID_0_0_0_MCHBAR (P_CR_CAPABILITY_ID_0_0_0_MCHBAR)—Offset 7018h" on page 1351 |
| 7020 | 4 | 00000000 | "MC_BIOS_REQ_0_0_0_MCH (P_CR_MC_BIOS_REQ_0_0_0_MCHBAR)—Offset 7020h" on page 1352 |
| 702C | 4 | 00000000 | "MCA_ERROR_SRC_0_0_0_MCHBAR (P_CR_MCA_ERROR_SRC_0_0_0_MCHBAR)—Offset 702Ch" on page 1353 |
| 7030 | 4 | 00010000 | "DDR_THERM_THRT_CTRL_0_0_0_MCHBAR (P_CR_DDR_THERM_THRT_CTRL_0_0_0_MCHBAR)—Offset 7030h" on page 1354 |
| 703C | 4 | 00000000 | "PACKAGE_THERM_MARGIN (P_CR_PACKAGE_THERM_MARGIN_0_0_0_MCHBAR)—Offset 703Ch" on page 1355 |
| 7040 | 8 | 0000000000000000 | "DDR_RAPL_LIMIT (P_CR_DDR_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 7040h" on page 1356 |
| 7048 | 4 | 00000000 | "DDR_ENERGY_STATUS (P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7048h" on page 1357 |
| 704C | 4 | 000058F0 | "DDR_RAPL_PERF_STATUS (P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 704Ch" on page 1357 |
| 7050 | 4 | 00000000 | "PACKAGE_RAPL_PERF_STATUS (P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 7050h" on page 1358 |
| 7054 | 4 | 00000000 | "PRIMARY_PLANE_TURBO_POWER_POLICY (P_CR_PRIMARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7054h" on page 1358 |
| 7058 | 4 | 00000010 | "SECONDARY_PLANE_TURBO_POWER_POLICY (P_CR_SECONDARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7058h" on page 1359 |
| 705C | 4 | 00000000 | "PRIMARY_PLANE_ENERGY_STATUS (P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 705Ch" on page 1359 |
| 7060 | 4 | 00000000 | "SECONDARY_PLANE_ENERGY_STATUS (P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7060h" on page 1360 |
| 7068 | 4 | 330A0E08 | "PACKAGE_POWER_SKU_UNIT (P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR)—Offset 7068h" on page 1361 |
| 706C | 4 | 00000000 | "PACKAGE_ENERGY_STATUS (P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 706Ch" on page 1362 |
| 7070 | 4 | 00000000 | "GT_PERF_STATUS (P_CR_GT_PERF_STATUS_0_0_0_MCHBAR)—Offset 7070h" on page 1362 |
| 7074 | 4 | 005A0000 | "TEMPERATURE_TARGET (P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR)—Offset 7074h" on page 1363 |
| 7078 | 4 | 00000000 | "BIOS_RESET_CPL (P_CR_BIOS_RESET_CPL_0_0_0_MCHBAR)—Offset 7078h" on page 1364 |
| 7080 | 4 | 00000000 | "BIOS_MAILBOX_DATA (P_CR_BIOS_MAILBOX_DATA_0_0_0_MCHBAR)—Offset 7080h" on page 1366 |
| 7084 | 4 | 00000000 | "BIOS_MAILBOX_INTERFACE (P_CR_BIOS_MAILBOX_INTERFACE_0_0_0_MCHBAR)—Offset 7084h" on page 1366 |

## Table 44-5. Summary of Memory Mapped I/O Registers for P-Unit—MCHBAR (Sheet 2 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 7088 | 4 | 00000000 | "CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7088h" on page 1367 |
| 708C | 4 | 00000000 | "GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 708Ch" on page 1368 |
| 7090 | 4 | 00000000 | "SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7090h" on page 1369 |
| 7094 | 4 | 00000000 | "FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7094h" on page 1369 |
| 7098 | 4 | 00000000 | "NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7098h" on page 1370 |
| 70A0 | 8 | 0012024000600118 | "PACKAGE_POWER_SKU (P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR)—Offset 70A0h" on page 1371 |
| 70A8 | 8 | 0000000000000000 | "PACKAGE_RAPL_LIMIT (P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 70A8h" on page 1372 |
| 70B0 | 4 | 00000000 | "IA_PERF_LIMIT_REASONS (P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR)—Offset 70B0h" on page 1374 |
| 70D8 | 4 | 00000000 | "MEMHOT_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_STATUS_0_0_0_MCHBAR)—Offset 70D8h" on page 1376 |
| 70DC | 4 | 00000000 | "MEMHOT_THERM_CONFIG_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_CONFIG_0_0_0_MCHBAR)—Offset 70DCh" on page 1376 |
| 7200 | 4 | 00000000 | "MEM_THERM_CTRL_0_0_0_MCHBAR (P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR)—Offset 7200h" on page 1377 |
| 7204 | 4 | 00645D54 | "MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR)—Offset 7204h" on page 1378 |
| 7208 | 4 | 00635A00 | "MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR)—Offset 7208h" on page 1378 |
| 720C | 4 | 00000000 | "MEM_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR)—Offset 720Ch" on page 1379 |
| 7210 | 4 | 00000000 | "MEM_THERM_INTERRUPT_0_0_0_MCHBAR (P_CR_MEM_THERM_INTERRUPT_0_0_0_MCHBAR)—Offset 7210h" on page 1380 |
| 7214 | 4 | 00000000 | "MEM_THERM_INT_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_INT_STATUS_0_0_0_MCHBAR)—Offset 7214h" on page 1381 |
| 7218 | 4 | 00000000 | "MEM_THERM_TEMP_OFST_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_OFST_0_0_0_MCHBAR)—Offset 7218h" on page 1383 |
| 70E0 | 8 | 0000000000000000 | "TELEM_NEAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_NEAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E0h" on page 1384 |
| 70E8 | 8 | 0000000000000000 | "TELEM_FAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_FAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E8h" on page 1384 |

## Table 44-5. Summary of Memory Mapped I/O Registers for P-Unit—MCHBAR (Sheet 3 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 70F0 | 4 | 00000000 | "TURBO_ACTIVATION_RATIO (P_CR_TURBO_ACTIVATION_RATIO_0_0_0_MCHBAR)—Offset 70F0h" on page 1385 |
| 7034 | 4 | 00000032 | "C2C3TT_CFG (P_CR_C2C3TT_CFG_0_0_0_MCHBAR)—Offset 7034h" on page 1385 |
| 7110 | 4 | 00000000 | "DTS_CONFIG1_CFG (P_CR_DTS_CONFIG1_CFG_0_0_0_MCHBAR)—Offset 7110h" on page 1386 |
| 7114 | 4 | 00000000 | "DTS_CONFIG2_CFG (P_CR_DTS_CONFIG2_CFG_0_0_0_MCHBAR)—Offset 7114h" on page 1386 |
| 7118 | 4 | 00000000 | "DTS_CONFIG3_CFG (P_CR_DTS_CONFIG3_CFG_0_0_0_MCHBAR)—Offset 7118h" on page 1387 |
| 711C | 4 | 00000000 | "SOUTH_IO_PORT_STATUS3 (P_CR_SOUTH_IO_PORT_STATUS3_0_0_0_MCHBAR)—Offset 711Ch" on page 1387 |
| 7128 | 4 | 00000000 | "DRAM_BIOS_INFO (P_CR_DRAM_BIOS_INFO_0_0_0_MCHBAR)—Offset 7128h" on page 1388 |
| 712C | 4 | 00000000 | "ADR_COMMAND (P_CR_ADR_COMMAND_0_0_0_MCHBAR)—Offset 712Ch" on page 1389 |
| 7130 | 8 | 0028000000000118 | "DRAM_POWER_INFO (P_CR_DRAM_POWER_INFO_0_0_0_MCHBAR)—Offset 7130h" on page 1390 |
| 7138 | 4 | 00000000 | "PCODE_WRITE_SPARE (P_CR_PCODE_WRITE_SPARE_0_0_0_MCHBAR)—Offset 7138h" on page 1390 |
| 70F4 | 4 | 00000000 | "PACKAGE_TEMPERATURES_0_0_0_MCHBAR (P_CR_PACKAGE_TEMPERATURES_0_0_0_MCHBAR)—Offset 70F4h" on page 1391 |
| 7108 | 4 | 00000000 | "MEMSS_FREQUENCY_CAPABILITIES (P_CR_MEMSS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7108h" on page 1392 |
| 7160 | 4 | 00000000 | "PP1_C0_CORE_CLOCK_0_0_0_MCHBAR (P_CR_PP1_C0_CORE_CLOCK_0_0_0_MCHBAR)—Offset 7160h" on page 1392 |
| 7164 | 4 | 00000000 | "Core Exists Vector (P_CR_CORE_EXISTS_VECTOR_0_0_0_MCHBAR)—Offset 7164h" on page 1393 |
| 7168 | 4 | 00000000 | "Software Core Disable Mask (P_CR_CORE_DISABLE_MASK_0_0_0_MCHBAR)—Offset 7168h" on page 1394 |
| 71F0 | 8 | 0000000000000000 | "PL3_CONTROL_MCHBAR (P_CR_PL3_CONTROL_0_0_0_MCHBAR)—Offset 71F0h" on page 1396 |
| 7220 | 8 | 0000000000000000 | "CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL1_0_0_0_MCHBAR)—Offset 7220h" on page 1398 |
| 7228 | 8 | 0000000000000000 | "CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL2_0_0_0_MCHBAR)—Offset 7228h" on page 1399 |
| 7230 | 8 | 0000000000000000 | "CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL3_0_0_0_MCHBAR)—Offset 7230h" on page 1400 |
| 7238 | 4 | 00000000 | "CONFIG_TDP_CONTROL (P_CR_CONFIG_TDP_CONTROL_0_0_0_MCHBAR)—Offset 7238h" on page 1400 |
| 723C | 4 | 00000000 | "CONFIG_TDP_NOMINAL (P_CR_CONFIG_TDP_NOMINAL_0_0_0_MCHBAR)—Offset 723Ch" on page 1401 |
| 7244 | 4 | 00000000 | "Graphics Superqueue Active Clocks (P_CR_PP1_ANY_THREAD_ACTIVITY_0_0_0_MCHBAR)—Offset 7244h" on page 1402 |

### 44.1.3 Host Memory Space—DEFVTDBAR

**Table 44-6. Summary of Memory Mapped I/O Registers—DEFVTDBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 64 | 4 | 00000000 | "B-Unit Copy of Default VTd BAR PMEN (B_CR_PMEN_REG_0_0_0_DEFVTDBAR)—Offset 64h" on page 1402 |
| 68 | 4 | 00000000 | "B-Unit Copy of Default VTd BAR PLM Base Register (B_CR_PLMBASE_REG_0_0_0_DEFVTDBAR)—Offset 68h" on page 1403 |
| 6C | 4 | 00000000 | "B-Unit Copy of Default VTd BAR PLM Limit Register (B_CR_PLMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 6Ch" on page 1404 |
| 70 | 8 | 0000000000000000 | "B-Unit Copy of Default VTd BAR PHM Base Register (B_CR_PHMBASE_REG_0_0_0_DEFVTDBAR)—Offset 70h" on page 1405 |
| 78 | 8 | 0000000000000000 | "B-Unit Copy of Default VTd BAR PHM Limit Register (B_CR_PHMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 78h" on page 1406 |

### 44.1.4 Host Memory Space—DEFVTDBAR

**Table 44-7. Summary of Memory Mapped I/O Registers—DEFVTDBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000010 | "Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h" on page 1407 |
| 8 | 8 | 00D2008C40660462 | "Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h" on page 1407 |
| 10 | 8 | 0000000000F050DA | "Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h" on page 1409 |
| 18 | 4 | 00000000 | "Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h" on page 1412 |
| 1C | 4 | 00000000 | "Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch" on page 1413 |
| 20 | 8 | 0000000000000000 | "Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h" on page 1414 |
| 28 | 8 | 0800000000000000 | "Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h" on page 1415 |
| 34 | 4 | 00000000 | "Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h" on page 1417 |
| 38 | 4 | 80000000 | "Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h" on page 1419 |
| 3C | 4 | 00000000 | "Fault Event Data Register (FEDATA_REG_0_0_0_VTDBAR)—Offset 3Ch" on page 1420 |
| 40 | 4 | 00000000 | "Fault Event Address Register (FEADDR_REG_0_0_0_VTDBAR)—Offset 40h" on page 1420 |
| 44 | 4 | 00000000 | "Fault Event Upper Address Register (FEUADDR_REG_0_0_0_VTDBAR)—Offset 44h" on page 1420 |
| 58 | 8 | 0000000000000000 | "Advanced Fault Log Register (AFLOG_REG_0_0_0_VTDBAR)—Offset 58h" on page 1421 |
| 64 | 4 | 00000000 | "Protected Memory Enable Register (PMEN_REG_0_0_0_VTDBAR)—Offset 64h" on page 1422 |
| 68 | 4 | 00000000 | "Protected Low Memory Base Register (PLMBASE_REG_0_0_0_VTDBAR)—Offset 68h" on page 1423 |
| 6C | 4 | 00000000 | "Protected Low-Memory Limit Register (PLMLIMIT_REG_0_0_0_VTDBAR)—Offset 6Ch" on page 1424 |
| 70 | 8 | 0000000000000000 | "Protected High-Memory Base Register (PHMBASE_REG_0_0_0_VTDBAR)—Offset 70h" on page 1425 |
| 78 | 8 | 0000000000000000 | "Protected High-Memory Limit Register (PHMLIMIT_REG_0_0_0_VTDBAR)—Offset 78h" on page 1426 |
| 80 | 8 | 0000000000000000 | "Invalidation Queue Head Register (IQH_REG_0_0_0_VTDBAR)—Offset 80h" on page 1427 |
| 88 | 8 | 0000000000000000 | "Invalidation Queue Tail Register (IQT_REG_0_0_0_VTDBAR)—Offset 88h" on page 1427 |
| 90 | 8 | 0000000000000000 | "Invalidation Queue Address Register (IQA_REG_0_0_0_VTDBAR)—Offset 90h" on page 1428 |
| 9C | 4 | 00000000 | "Invalidation Completion Status Register (ICS_REG_0_0_0_VTDBAR)—Offset 9Ch" on page 1428 |
| A0 | 4 | 80000000 | "Invalidation Event Control Register (IECTL_REG_0_0_0_VTDBAR)—Offset A0h" on page 1429 |
| A4 | 4 | 00000000 | "Invalidation Event Data Register (IEDATA_REG_0_0_0_VTDBAR)—Offset A4h" on page 1430 |
| A8 | 4 | 00000000 | "Invalidation Event Address Register (IEADDR_REG_0_0_0_VTDBAR)—Offset A8h" on page 1430 |

## Table 44-7. Summary of Memory Mapped I/O Registers—DEFVTDBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| AC | 4 | 00000000 | "Invalidation Event Upper Address Register (IEUADDR_REG_0_0_0_VTDBAR)—Offset ACh" on page 1431 |
| B8 | 8 | 0000000000000000 | "Interrupt Remapping Table Address Register (IRTA_REG_0_0_0_VTDBAR)—Offset B8h" on page 1431 |
| DC | 4 | 00000000 | "Page Request Status Register (PRESTS_REG_0_0_0_VTDBAR)—Offset DCh" on page 1432 |
| E0 | 4 | 80000000 | "Page Request Event Control Register (PRECTL_REG_0_0_0_VTDBAR)—Offset E0h" on page 1433 |
| E4 | 4 | 00000000 | "Page Request Event Data Register (PREDATA_REG_0_0_0_VTDBAR)—Offset E4h" on page 1434 |
| E8 | 4 | 00000000 | "Page Request Event Address Register (PREADDR_REG_0_0_0_VTDBAR)—Offset E8h" on page 1434 |
| EC | 4 | 00000000 | "Page Request Event Upper Address Register (PREUADDR_REG_0_0_0_VTDBAR)—Offset ECh" on page 1434 |
| 400 | 8 | 0000000000000000 | "Fault Recording Register Low [0] (FRCDL_REG_0_0_0_VTDBAR)—Offset 400h" on page 1435 |
| 408 | 8 | 0000000000000000 | "Fault Recording Register High [0] (FRCDH_REG_0_0_0_VTDBAR)—Offset 408h" on page 1436 |
| 500 | 8 | 0000000000000000 | "Invalidate Address Register (IVA_REG_0_0_0_VTDBAR)—Offset 500h" on page 1438 |
| 508 | 8 | 0200000000000000 | "IOTLB Invalidate Register (IOTLB_REG_0_0_0_VTDBAR)—Offset 508h" on page 1439 |

### 44.1.5 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (BAR) DEFVTDBAR. It is located in Host Root Memory Space MCHBAR +offset 6C80h.

See Section 44.3.123, "Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR)—Offset 6C80h" on page 1316. Also see Section 3.4.2, "Accessing System Agent Sideband Registers" on page 187.

The Sideband registers for Memory Controller 0 are shown in Table 44-8. For Memory Controller 1 see Table 44-9.

**Table 44-8. Summary of Sideband Registers—0x10 (Memory Controller 0)**

| Offset from Sideband Port 10h (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 64 | 10_0064 | 00000800 | "UCELOG—Offset 64h" on page 1441 |
| 68 | 10_0068 | 00000000 | "UCEADDR—Offset 68h" on page 1442 |
| 6C | 10_006C | 00000800 | "SBELOG—Offset 6Ch" on page 1442 |
| 70 | 10_0070 | 00000000 | "SBEADDR—Offset 70h" on page 1443 |
| 74 | 10_0074 | 00000000 | "SBECNT0—Offset 74h" on page 1444 |
| 78 | 10_0078 | 00000000 | "SBECNT1—Offset 78h" on page 1444 |
| 7C | 10_007C | 00000000 | "SBECNT2—Offset 7Ch" on page 1445 |
| 80 | 10_0080 | 00000000 | "SBECNT3—Offset 80h" on page 1445 |
| 84 | 10_0084 | 00000000 | "SBEACC0—Offset 84h" on page 1445 |
| 88 | 10_0088 | 00000000 | "SBEACC1—Offset 88h" on page 1446 |
| 8C | 10_008C | 00000000 | "SBEACC2—Offset 8Ch" on page 1446 |
| 90 | 10_0090 | 00000000 | "SBEACC3—Offset 90h" on page 1446 |
| B0 | 10_00B0 | 00000000 | "DPATROL_SCRUB_CFG—Offset B0h" on page 1447 |
| CC | 10_00CC | 00000000 | "UCELOG1—Offset CCh" on page 1447 |
| D0 | 10_00D0 | 00000000 | "SBELOG1—Offset D0h" on page 1448 |
| 100 | 10_0100 | 00000000 | "CORRERRTHRSHLD0—Offset 100h" on page 1448 |
| 104 | 10_0104 | 00000000 | "CORRERRTHRSHLD1—Offset 104h" on page 1448 |
| 108 | 10_0108 | 00000000 | "CORRERRTHRSHLD2—Offset 108h" on page 1449 |
| 10C | 10_010C | 00000000 | "CORRERRTHRSHLD3—Offset 10Ch" on page 1449 |
| 110 | 10_0110 | 00000000 | "LEAKY_BUCKET_CFG0—Offset 110h" on page 1449 |
| 114 | 10_0114 | 00000000 | "LEAKY_BUCKET_CFG1—Offset 114h" on page 1450 |
| 118 | 10_0118 | 00000000 | "LEAKY_BUCKET_CFG2—Offset 118h" on page 1450 |
| 11C | 10_011C | 00000000 | "LEAKY_BUCKET_CFG3—Offset 11Ch" on page 1450 |
| 120 | 10_0120 | 00000000 | "FERRNERR—Offset 120h" on page 1451 |
| 124 | 10_0124 | 00000000 | "DERRSTS—Offset 124h" on page 1452 |
| 128 | 10_0128 | 00000000 | "DERRMSKSEV—Offset 128h" on page 1453 |
| 12C | 10_012C | 00000000 | "DERRCNTSEL—Offset 12Ch" on page 1454 |
| 130 | 10_0130 | 00000000 | "DERRCNTSEL—Offset 12Ch" on page 1454 |
| 134 | 10_0134 | 00000000 | "ERRINJCTL—Offset 134h" on page 1455 |

## Table 44-8. Summary of Sideband Registers—0x10 (Continued) (Memory Controller 0)

| Offset from Sideband Port 10h (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 138 | 10_0138 | 00000000 | "ERRINJADDR—Offset 138h" on page 1456 |
| 13C | 10_013C | 00000000 | "ERRINJAMSK—Offset 13Ch" on page 1456 |
| 140 | 10_0140 | 00000000 | "ERRINJDATA0—Offset 140h" on page 1456 |
| 144 | 10_0144 | 00000000 | "ERRINJDATA1—Offset 144h" on page 1456 |
| 170 | 10_0170 | 00000000 | "DCS—Offset 170h" on page 1457 |
| 180 | 10_0180 | 00000000 | "DECCCTRL—Offset 180h" on page 1458 |
| 18C | 10_018C | 00000000 | "LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch" on page 1459 |
| 194 | 10_0194 | FFFFFFFF | "DPATROL_SCRUB_TMR—Offset 194h" on page 1459 |

## Table 44-9. Summary of Sideband Registers—0x12 (Memory Controller 1)

| Offset from Sideband Port 12h (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 64 | 12_0064 | 00000800 | "UCELOG—Offset 64h" on page 1460 |
| 68 | 12_0068 | 00000000 | "UCEADDR—Offset 68h" on page 1460 |
| 6C | 12_006C | 00000800 | "SBELOG—Offset 6Ch" on page 1461 |
| 70 | 12_0070 | 00000000 | "SBEADDR—Offset 70h" on page 1461 |
| 74 | 12_0074 | 00000000 | "SBECNT0—Offset 74h" on page 1462 |
| 78 | 12_0078 | 00000000 | "SBECNT1—Offset 78h" on page 1462 |
| 7C | 12_007C | 00000000 | "SBECNT2—Offset 7Ch" on page 1463 |
| 80 | 12_0080 | 00000000 | "SBECNT3—Offset 80h" on page 1463 |
| 84 | 12_0084 | 00000000 | "SBEACC0—Offset 84h" on page 1464 |
| 88 | 12_0088 | 00000000 | "SBEACC1—Offset 88h" on page 1464 |
| 8C | 12_008C | 00000000 | "SBEACC2—Offset 8Ch" on page 1465 |
| 90 | 12_0090 | 00000000 | "SBEACC3—Offset 90h" on page 1465 |
| B0 | 12_00B0 | 00000000 | "DPATROL_SCRUB_CFG—Offset B0h" on page 1466 |
| CC | 12_00CC | 00000000 | "UCELOG1—Offset CCh" on page 1466 |
| D0 | 12_00D0 | 00000000 | "SBELOG1—Offset D0h" on page 1467 |
| 100 | 12_0100 | 00000000 | "CORRERRTHRSHLD0—Offset 100h" on page 1467 |
| 104 | 12_0104 | 00000000 | "CORRERRTHRSHLD1—Offset 104h" on page 1467 |
| 108 | 12_0108 | 00000000 | "CORRERRTHRSHLD2—Offset 108h" on page 1468 |
| 10C | 12_010C | 00000000 | "CORRERRTHRSHLD3—Offset 10Ch" on page 1468 |
| 110 | 12_0110 | 00000000 | "LEAKY_BUCKET_CFG0—Offset 110h" on page 1468 |
| 114 | 12_0114 | 00000000 | "LEAKY_BUCKET_CFG1—Offset 114h" on page 1469 |
| 118 | 12_0118 | 00000000 | "LEAKY_BUCKET_CFG2—Offset 118h" on page 1469 |
| 11C | 12_011C | 00000000 | "LEAKY_BUCKET_CFG3—Offset 11Ch" on page 1469 |
| 120 | 12_0120 | 00000000 | "FERRNERR—Offset 120h" on page 1470 |
| 124 | 12_0124 | 00000000 | "DERRSTS—Offset 124h" on page 1471 |
| 128 | 12_0128 | 00000000 | "DERRMSKSEV—Offset 128h" on page 1472 |
| 12C | 12_012C | 00000000 | "DERRCNTSEL—Offset 12Ch" on page 1473 |
| 130 | 12_0130 | 00000000 | "DERRCNT—Offset 130h" on page 1473 |
| 134 | 12_0134 | 00000000 | "ERRINJCTL—Offset 134h" on page 1474 |
| 138 | 12_0138 | 00000000 | "ERRINJADDR—Offset 138h" on page 1475 |
| 13C | 12_013C | 00000000 | "ERRINJAMSK—Offset 13Ch" on page 1475 |
| 140 | 12_0140 | 00000000 | "ERRINJDATA0—Offset 140h" on page 1475 |
| 144 | 12_0144 | 00000000 | "ERRINJDATA1—Offset 144h" on page 1475 |
| 170 | 12_0170 | 00000000 | "DCS—Offset 170h" on page 1476 |
| 180 | 12_0180 | 00000000 | "DECCCTRL—Offset 180h" on page 1478 |
| 18C | 12_018C | 00000000 | "LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch" on page 1479 |
| 194 | 12_0194 | FFFFFFFF | "DPATROL_SCRUB_TMR—Offset 194h" on page 1479 |

## Table 44-10. Summary of Sideband Registers—0x99

| Offset from Sideband Port 99h (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 220 | 99_0220 | 00042222 | "ICLK_DWORD8—Port 99h, Offset 220h" on page 1480 |

## 44.2 Registers in Configuration Space

### 44.2.1 Device ID and Vendor ID Register (DEVICE_ID_VENDOR_ID_0_0_0_PCI)—Offset 0h

This register uniquely identifies any PCI device.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 0h

**Default:** 19908086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x1990<br>RO | **Device ID (DID):** This field identifies the particular function as allocated by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VENDOR_ID):** This field identifies Intel as the manufacturer of the device. |

### 44.2.2 PCI Status and PCI Command Register (PCI_STATUS_COMMAND_0_0_0_PCI)—Offset 4h

PCI Status is used to record status information for PCI bus related events. PCI Command provides coarse control over a device's ability to generate and respond to PCI cycles.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 4h

**Default:** 00000007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x7<br>RO | **PCI Status and PCI Command (PCI_STATUS_AND_COMMAND):**<br>Allowing this device to respond to I/O space and Memory Space accesses.<br>This device is also allowed to behave as a bus master. |

### 44.2.3 PCI Revision ID and PCI Class Code Register (REVISION_ID_CLASS_CODE_0_0_0_PCI)—Offset 8h

Revision ID contains the revision number of the device. Class Code identifies the basic function of the device.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 8h

**Default:** 06000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x600<br>RO | **Class Code 1 (CLASS_CODE1):** This device is a Host Bridge. |
| 15:8 | 0x0<br>RO | **Class Code 0 (CLASS_CODE0):** This device is a Host Bridge. |
| 7:0 | 0x0<br>RW/V | **Revision ID (REVISION_ID):** Revision ID |

### 44.2.4 Master Latency Timer and Header Type Register (MASTER_LATENCY_TIME_0_0_0_PCI)—Offset Ch

This register defines Latency Timer and layout of the device Configuration Space header.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 23:16 | 0x80<br>RO | **Header Type (HEADER_TYPE):** This is a multi-function device. |
| 15:8 | 0x0<br>RO | **Master Latency Timer (MASTER_LATENCY_TIMER):** This field specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. |
| 7:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

## 44.2.5 PCI Subsystem Vendor ID and PCI Subsystem ID (SVID_SID_0_0_0_PCI)—Offset 2Ch

This register is used to uniquely identify the subsystem where the PCI device resides.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Subsystem ID (SUBSYSTEM_ID):** PCI Subsystem ID:<br>This field should be programmed during BIOS initialization. |
| 15:0 | 0x0<br>RW | **Subsystem Vendor ID (SUBSYSTEM_VENDOR_ID):** PCI Subsystem Vendor ID:<br>This field should be programmed by BIOS during boot-up to indicate the vendor of the system board. |

## 44.2.6 Capability Register Pointer (CAPPTR_0_0_0_PCI)—Offset 34h

This register contains a pointer to the first Capability Register in a linked list.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 34h

**Default:** 000000E0h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 7:0 | 0xe0<br>RO | **Base Address of First Capability Register (BASE_ADDR):** This pointer is an 8-bit address to an offset within this device's Configuration Space that holds the first Capability Register (CAPID0_CAPCTRL). |

## 44.2.7 Memory Controller Hub Base Address Register (MCHBAR_LO_0_0_0_PCI)—Offset 48h

This register contains the lower 32bits of the MCHBAR base address.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:15 | 0x0<br>RW | **Base Address of MCHBAR (BASE_ADDR):** Defines the base address of MCHBAR.<br>MCHBAR[38:15] is MCHBAR_HI[6:0],MCHBAR_LO[31:15].<br>If incoming Request Address[38:15] matches MCHBAR[38:15] then the request hits and address in the MCHBAR range. |
| 14:4 | 0x0<br>RO | **Reserved:** Reserved |
| 3:1 | 0x0<br>RO | **Reserved:** Reserved |
| 0 | 0x0<br>RW | **MCHBAR Enable (MCHBAREN):**<br>• 0: MCHBAR is disabled and does not claim any memory<br>• 1: MCHBAR memory mapped accesses are claimed and decoded appropriately |

## 44.2.8 Memory Controller Hub Base Address Register (MCHBAR_HI_0_0_0_PCI)—Offset 4Ch

This register contains the upper 32bits of the MCHBAR base address.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 6:0 | 0x0<br>RW | **Base Address of MCHBAR (BASE_ADDR):** Defines the base address of the MCHBAR.<br>If incoming Request Address[38:15] matches MCHBAR[38:15], then request hits an address in the MCHBAR range. |

## 44.2.9 Device Enable Register (DEVEN_0_0_0_PCI)—Offset 54h

The DEVEN register allows for enabling/disabling of PCI devices and functions that are within the CPU package.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 54h

**Default:** 00000033h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 5 | 0x1<br>RW/V | **Device 3 Function 0 Enable (D3F0EN):**<br>• 0: Bus 0 Device 3 Function 0 is disabled and hidden<br>• 1: Bus 0 Device 3 Function 0 is enabled and visible<br>This bit will be set to 0b and remain 0b if Device 3 capability is disabled. |
| 4 | 0x1<br>RW/V | **Device 2 Function 0 Enable (D2F0EN):**<br>• 0: Bus 0 Device 2 Function 0 is disabled and hidden<br>• 1: Bus 0 Device 2 Function 0 is enabled and visible<br>This bit will be set to 0b and remain 0b if Device 2 capability is disabled. |
| 3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 1 | 0x1<br>RW/V | **Device 0 Function 1 Enable (D0F1EN):**<br>• 0: Bus 0 Device 0 Function 1 is disabled and hidden<br>• 1: Bus 0 Device 0 Function 1 is enabled and visible<br>This bit will be set to 0b and remain 0b if Device 0/0/1 capability is disabled. |
| 0 | 0x1<br>RO | **Device 0 Function 0 Enable (D0F0EN):** Bus 0 Device 0 Function 0 may not be disabled, and is therefore hardwired to 1. |

## 44.2.10 PCI Express Enhanced Configuration Range Base Address Low (PCIEXBAR_LO_0_0_0_PCI)—Offset 60h

This register contains the lower 32bits of the base address of the PCI Express Enhanced Configuration region.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **PCI Express Base Address Register (PCIEXBAR):** PCIEXBAR[38:28] is PCIEXBAR_HI[6:0],PCIEXBAR_LO[31:28].<br>Describes bits [38:28] of the base address of the contiguous 256MB region for PCI Express Enhanced Configuration region.<br>If bits [38:28] of the request address matches the PCIEXBAR[38:28] then the request targets the PCI Express Enhanced Configuration Space region.<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Address Mask for 128MB region (ADMSK128):** Reserved.<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Address Mask for 64MB region (ADMSK64):** Reserved.<br><br>**Power Well:** prst |
| 25:3 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 2:1 | 0x0<br>RO | **Length of the Region (LENGTH):** Reserved.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **PCIEXBAR Range Enable (PCIEXBAREN):**<br>• 0: The PCIEXBAR range is disabled. Address may target DRAM or MMIO, depending on other address decode rules.<br>• 1: The PCIEXBAR range is enabled. Incoming request address must be compared with PCIEXBAR to determine whether the request targets PCI Express Enhanced Configuration region.<br><br>**Power Well:** prst |

## 44.2.11 PCI Express Enhanced Configuration Range Base Address High (PCIEXBAR_HI_0_0_0_PCI)—Offset 64h

This register contains the upper 32bits of the base address of the PCI Express Enhanced Configuration region.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 6:0 | 0x0<br>RW | **PCI Express Base Address Register (PCIEXBAR_HI):** PCIEXBAR[38:28] is PCIEXBAR_HI[6:0],PCIEXBAR_LO[31:28].<br>Describes bits [38:28] of the base address of the contiguous 256MB region for PCI Express Enhanced Configuration region.<br>If bits [38:28] of the request address matches the PCIEXBAR[38:28] then the request targets the PCI Express Enhanced Configuration Space region.<br><br>**Power Well:** prst |

## 44.2.12 Top of Upper Usable DRAM Low (TOUUD_LO_0_0_0_PCI)—Offset A8h

Defines the top of the upper usable DRAM range and start of the upper MMIO address range. Formerly defined in BMBOUND_HI. This register contains the lower 32 bits of TOUUD.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + A8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Top of Upper Usable DRAM (TOUUD):** TOUUD[38:20] is TOUUD_HI[6:0],TOUUD_LO[31:20].<br>Upper usable DRAM address range ends at the preceding byte.<br>Upper MMIO Range starts at this address and extends up to the maximum system addressable memory range.<br>Bits 38:20 are compared with bits 38:20 of the incoming request address to determine whether the request targets the upper usable DRAM range or upper MMIO range.<br>If Request Address[38:20])=TOUUD[38:20] and any bit in Request Address[38:32] is set, request is determined to target the upper MMIO range.<br><br>**Power Well:** prst |
| 19:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved should return 0s on read.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **TOUUD Register Lock (LOCK_RESERVED):** This bit will lock all writable settings in this register, including itself (client definition).<br><br>**Power Well:** prst |

## 44.2.13 Top of Upper Usable DRAM High (TOUUD_HI_0_0_0_PCI)—Offset ACh

Defines the top of the upper usable DRAM range and start of the upper MMIO address range. Formerly defined in BMBOUND_HI. This register contains the upper 32 bits of TOUUD.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + ACh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 6:0 | 0x0<br>RW | **Top of Upper Usable DRAM (TOUUD_HI):** TOUUD[38:20] is TOUUD_HI[6:0],TOUUD_LO[31:20].<br>Upper usable DRAM address range ends at the preceding byte.<br>Upper MMIO Range starts at this address and extends up to the maximum system addressable memory range.<br>Bits 38:20 are compared with bits 38:20 of the incoming request address to determine whether the request targets the upper usable DRAM range or upper MMIO range.<br>If Request Address[38:20])=TOUUD[38:20] and any bit in Request Address[38:32] is set, request is determined to target the upper MMIO range.<br><br>**Power Well:** prst |

## 44.2.14 Base of Data Stolen Memory (BDSM_0_0_0_PCI)—Offset B0h

Defines the Base of the Data Stolen Memory. In addition, the GSM range is defined to end at the preceding byte. See BGSM register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Base of Data Stolen Memory (BDSM):** This register contains the base address of the Data Stolen Memory.<br>The limit for the Data Stolen Memory is TOLUD-1.<br>This range is not decoded by the system agent, but is a sub-region of BGSM decoded by the Integrated Graphics Device.<br>Incoming Request Address[31:20] is compared against BDSM[31:20] and TOLUD[31:20] to determine if the address falls in the range.<br>The comparison check is as follows: BDSM[31:20](=Address[31:20] and Address[31:20](TOLUD[31:20] and Address[38:32]==0.<br>Request SAI is then checked against the allowed SAIs to determine if access is allowed.<br><br>**Power Well:** prst |
| 19:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **BDSM Register Lock (LOCK):** This bit will lock all writable settings in this register, including itself.<br><br>**Power Well:** prst |

## 44.2.15 Base of Graphics Stolen Memory (BGSM_0_0_0_PCI)—Offset B4h

Defines the Base of the Graphics Stolen Memory. In addition, the SMM range is defined to end at the preceding byte. See TSEGMB register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + B4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Base of Graphics Stolen Memory (BGSM):** This register contains the base address of the Graphics Stolen Memory.<br>The limit for the Graphics Stolen Memory is TOLUD-1.<br>Incoming Request Address[31:20] is compared against BGSM[31:20] and TOLUD[31:20] to determine if the address falls in the range.<br><br>**Power Well:** prst |
| 19:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **BGSM Register Lock (LOCK):** This bit will lock all writable settings in this register, including itself.<br><br>**Power Well:** prst |

## 44.2.16 Top Segment Memory Base (TSEGMB_0_0_0_PCI)—Offset B8h

TSEGMB defines the base of the SMM range, and the BGSM register defines the limit of the SMM range.

*Note:* In prior SoCs, the base register was defined in SMMRRL register and the limit was defined in the SMMRRH register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + B8h

**Default:** 00100000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x1<br>RW | **Base address of Top Segment DRAM Memory (TSEGMB):** BIOS determines the base of TSEG memory, which must be at or below Graphics Base of GTT Stolen Memory BGSM.<br>SMM range starts at this base and ends at BGSM1.<br>Incoming Request Address [31:20] will be compared with TSEGMB[31:20] and BGSM[31:20] to determine if the request targets the SMM range.<br><br>**Power Well:** prst |
| 19:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **TSEGMB Register Lock (LOCK):** This bit will lock all writable settings in this register, including itself.<br><br>**Power Well:** prst |

## 44.2.17 Top of Lower Usable DRAM (TOLUD_0_0_0_PCI)—Offset BCh

This register defines the Top of Lower Usable DRAM range and start of the Lower MMIO Address range. Formerly defined in BMBOUND.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + BCh

**Default:** 00100000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x1<br>RW | **Top of Lower Usable DRAM (TOLUD):** Defines the top of lower usable DRAM, which ends at the preceding byte.<br>Lower MMIO Address range starts at this address and continues up to the 4GB Address 0xFFFF_FFFF.<br>Bits 31:20 are compared with incoming request Address[31:20] to determine whether the request targets lower usable DRAM range or the lower MMIO range.<br>If Request Address[31:20])=TOLUD[31:20] and Request Address[38:32]==0 then the Request Address falls in the Lower MMIO Address range.<br><br>**Power Well:** prst |
| 19:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **TOLUD register lock (LOCK):** Reserved.<br><br>**Power Well:** prst |

## 44.2.18 Scratchpad (SKPD_0_0_0_PCI)—Offset DCh

This register holds 32 writable bits with no functionality behind them. It is for the convenience of BIOS and graphics drivers.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + DCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Scratchpad (SKPD):** 1 DWORD of data storage. |

## 44.2.19 Capability ID0 Capability Control (CAPID0_CAPCTRL0_0_0_0_PCI)—Offset E0h

Control bits in this register describe the attributes of CAPID0_A and CAPID0_B capability registers.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + E0h

**Default:** 010C0009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 27:24 | 0x1<br>RO | **Capability ID Version (CAPID_VER):** This field has the value 0001b to identify the first revision of the CAPID register definition. |
| 23:16 | 0xc<br>RO | **Capability ID0 Structure Length (CAPIDLEN):** This field has the value 0Ch to indicate the structure length 12 bytes.<br>This is the total size of this CAPCTRL and the CAPID0_A and CAPID0_B registers in the following bytes. |
| 15:8 | 0x0<br>RO | **Next Capability Register Pointer (NEXT_CAP):** This field is hardwired to 00h, indicating the end of the capabilities linked list. |
| 7:0 | 0x9<br>RO | **Capability ID (CAP_ID):** This field has the value 1001b to identify the CAP_ID assigned by the PCI SIG for vendor dependent capability pointers. |

## 44.2.20 Capability ID0 A (CAPID0_A_0_0_0_PCI)—Offset E4h

Control of bits in this register is only required for customer visible SKU differentiation.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + E4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **Reserved:** Reserved |
| 23 | 0x0<br>RW/Fuse | **VTd Disable (VTDD):**<br>• 0: Enable VTd<br>• 1: Disable VTd |
| 22:16 | 0x0<br>RW/Fuse | **Reserved:** Reserved for future capabilities. |
| 15 | 0x0<br>RO/V | **Dynamic Power Performance Management (DPPM) Disable:**<br>• 0: DPPM associated memory spaces are accessible.<br>• 1: DPPM associated memory and IO spaces are disabled. DEVEN_0_0_0_PCI field for DPPM cannot be set. |
| 14:0 | 0x0<br>RW/Fuse | **Reserved:** Reserved |

## 44.2.21 Capability ID0 B (CAPID0_B_0_0_0_PCI)—Offset E8h

Control of bits in this register is only required for customer visible SKU differentiation.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + E8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Reserved:** Reserved |
| 30 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 29 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 28 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 27 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 26 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 25 | 0x0<br>RW/Fuse | **Reserved:** Reserved |
| 24 | 0x0<br>RO/V | **Shared Virtual Memory Disable (SVMDIS):**<br>• 0: Enable Shared Virtual Memory mode<br>• 1: Disable Shared Virtual Memory mode |
| 23:0 | 0x0<br>RW | **Spare bits[23:0] (SPARE23_0):** Reserved for future capabilities. |

## 44.2.22 Manufacturer ID Register (MANUFACTURER_ID_0_0_0_PCI)—Offset F4h

This register describes the manufacturing characteristics of the device.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:0, F:0] + F4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW/V | **Reserved (RESERVED_0):** Reserved. |
| 27:24 | 0x0<br>RW/V | **DOT Portion of Process ID (DOT_PORTION_OF_PROCESS_ID):** |
| 23:16 | 0x0<br>RW/V | **Process Portion or Process ID (PROCESS_PORTION_OF_PROCESS_ID):** |
| 15:8 | 0x0<br>RW/V | **Manufacturing ID (MID):** |
| 7:0 | 0x0<br>RW/V | **Manufacturing Stepping ID (MSID):** |

## 44.3 Registers in Memory Space—MCHBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) MCHBAR. It is a 64-bit BAR and is located in Host Root Configuration Space at Bus 0, Device 0, Function 0, offset 48h and 4Ch.

See Section 44.2.7, "Memory Controller Hub Base Address Register (MCHBAR_LO_0_0_0_PCI)—Offset 48h" on page 1106 and Section 44.2.8, "Memory Controller Hub Base Address Register (MCHBAR_HI_0_0_0_PCI)—Offset 4Ch" on page 1106.

### 44.3.1 Default VTd Base Address Register (DEFVTDBAR_0_0_0_MCHBAR_C)—Offset 6C80h

This is the base address for the Default VT configuration space. There is no physical memory within this 4KB window that can be addressed. The 4KB reserved by this register does not alias to any PCI 2.3 compliant memory mapped space. On reset the DEFVTd configuration space is disabled and must be enabled by writing a 1'b1 to DEFVTBAREN. BIOS programs this register after which the register cannot be altered. BIOS must write DEFVTDBAR and then immediately follow it up with a read to DEFVTDBAR to ensure that all copies of DEFVTDBAR in the system are updated.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6C80h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:40 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 39 | 0x0<br>RO | **Reserved:** Reserved |
| 38:12 | 0x0<br>RW | **Default VTd Base Address Register (DEFVTDBAR):** If DEFVTDBAR is enabled this field corresponds to bits 38:12 of the base address default IOMMU VTd configuration space.<br>BIOS will program this register, resulting in a base address for a 4KB block of contiguous memory address space.<br>This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space.<br>System Software uses this base address to program the default VTd IOMMU register set.<br>If DEFVTDBAR is enabled and incoming Request Address[38:12] matches DEFVTDBAR[38:12], the request targets the Default VTd BAR. |
| 11:2 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 1 | 0x0<br>RW | **DEFVTDBAR Register Lock (LOCK):** Locks the contents of the register, including itself. |
| 0 | 0x0<br>RW/L | **Default VTd Base Address Range Enable (DEFVTDBAREN):**<br>• 0: DEFVTDBAR is disabled and does not claim any memory<br>• 1: DEFVTDBAR memory mapped accesses are claimed and decoded appropriately.<br>This bit will remain 1'b0 if VTd capability is disabled. |

## 44.3.2 Spare BIOS (A_CR_SPARE_BIOS_MCHBAR)—Offset 647Ch

Spare CR in MCHBAR.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 647Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **SPARE RW Bits (SPARE_RW):** Spare RW 32 bits in BIOSWR policy group. |

## 44.3.3 Uncorrectable Error Status Register (A_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6588h

Errors that have been seen in A-Unit. The error is only logged if the respective bit in UNCERRMSK is 0.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6588h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RW/1C | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error<br><br>**Power Well:** prst |
| 24 | 0x0<br>RW/1C | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error<br><br>**Power Well:** prst |
| 23 | 0x0<br>RW/1C | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error<br><br>**Power Well:** prst |
| 22 | 0x0<br>RW/1C | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error<br><br>**Power Well:** prst |
| 21 | 0x0<br>RW/1C | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error<br><br>**Power Well:** prst |
| 20 | 0x0<br>RW/1C | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19 | 0x0<br>RW/1C | **downstream data buffer parity error (downstream_data_buffer_par_err):** downstream data buffer parity error detected<br><br>**Power Well:** prst |
| 18 | 0x0<br>RW/1C | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** upstream cmd buffer parity error detected<br><br>**Power Well:** prst |
| 17 | 0x0<br>RW/1C | **iosf upstream data parity error (iosf_upstream_data_par_err):** iosf upstream data parity error detected<br><br>**Power Well:** prst |
| 16 | 0x0<br>RW/1C | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** iosf upstream cmd parity error detected<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RW/1C | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set<br><br>**Power Well:** prst |
| 9 | 0x0<br>RW/1C | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large<br><br>**Power Well:** prst |
| 8 | 0x0<br>RW/1C | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction<br><br>**Power Well:** prst |
| 7 | 0x0<br>RW/1C | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **reserved (undefined5):** Reserved for future use.<br><br>**Power Well:** prst |
| 5 | 0x0<br>RW/1C | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction.<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW/1C | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf.<br><br>**Power Well:** prst |
| 3 | 0x0<br>RW/1C | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction.<br><br>**Power Well:** prst |
| 2 | 0x0<br>RW/1C | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf.<br><br>**Power Well:** prst |
| 1 | 0x0<br>RW/1C | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW/1C | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf.<br><br>**Power Well:** prst |

## 44.3.4 Uncorrectable Error Mask Register (A_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 658Ch

Masks whether a reported error is logged and signaled to the IEH.

1- do not log/signal.

0 - log/signal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 658Ch

**Default:** FFFFFFFFh

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Reserved** |
| 30 | 0x1<br>RO | **Reserved** |
| 29 | 0x1<br>RO | **Reserved** |
| 28 | 0x1<br>RO | **Reserved** |
| 27 | 0x1<br>RO | **Reserved** |
| 26 | 0x1<br>RO | **Reserved** |
| 25 | 0x1<br>RW | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error |
| 24 | 0x1<br>RW | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error |
| 23 | 0x1<br>RW | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error |
| 22 | 0x1<br>RW | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error |
| 21 | 0x1<br>RW | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error |
| 20 | 0x1<br>RW | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected |
| 19 | 0x1<br>RW | **downstream data buffer parity error (downstream_data_buffer_par_err):** downstream data buffer parity error detected |
| 18 | 0x1<br>RW | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** upstream cmd buffer parity error detected |
| 17 | 0x1<br>RW | **iosf upstream data parity error (iosf_upstream_data_par_err):** iosf upstream data parity error detected |
| 16 | 0x1<br>RW | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** iosf upstream cmd parity error detected |
| 15 | 0x1<br>RO | **Reserved** |
| 14 | 0x1<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 13 | 0x1<br>RO | **Reserved** |
| 12 | 0x1<br>RO | **Reserved** |
| 11 | 0x1<br>RO | **Reserved** |
| 10 | 0x1<br>RW | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set |
| 9 | 0x1<br>RW | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large |
| 8 | 0x1<br>RW | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction |
| 7 | 0x1<br>RW | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value |
| 6 | 0x1<br>RO | **Reserved** |
| 5 | 0x1<br>RW | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction. |
| 4 | 0x1<br>RW | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf. |
| 3 | 0x1<br>RW | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction. |
| 2 | 0x1<br>RW | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf. |
| 1 | 0x1<br>RW | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction. |
| 0 | 0x1<br>RW | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf. |

## 44.3.5 Uncorrectable Error Severity Register (A_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6590h

When set, escalate error as fatal to IEH, otherwise non-fatal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6590h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved** |
| 30 | 0x0<br>RO | **Reserved** |
| 29 | 0x0<br>RO | **Reserved** |
| 28 | 0x0<br>RO | **Reserved** |
| 27 | 0x0<br>RO | **Reserved** |
| 26 | 0x0<br>RO | **Reserved** |
| 25 | 0x0<br>RW | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error |
| 24 | 0x0<br>RW | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error |
| 23 | 0x0<br>RW | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error |
| 22 | 0x0<br>RW | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error |
| 21 | 0x0<br>RW | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error |
| 20 | 0x0<br>RW | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected |
| 19 | 0x0<br>RW | **downstream data buffer parity error (downstream_data_buffer_par_err):** downstream data buffer parity error detected |
| 18 | 0x0<br>RW | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** upstream cmd buffer parity error detected |
| 17 | 0x0<br>RW | **iosf upstream data parity error (iosf_upstream_data_par_err):** iosf upstream data parity error detected |
| 16 | 0x0<br>RW | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** iosf upstream cmd parity error detected |
| 15 | 0x0<br>RO | **Reserved** |
| 14 | 0x0<br>RO | **Reserved** |
| 13 | 0x0<br>RO | **Reserved** |
| 12 | 0x0<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x0<br>RO | **Reserved** |
| 10 | 0x0<br>RW | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set. |
| 9 | 0x0<br>RW | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large |
| 8 | 0x0<br>RW | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction |
| 7 | 0x0<br>RW | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value |
| 6 | 0x0<br>RO | **Reserved** |
| 5 | 0x0<br>RW | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction. |
| 4 | 0x0<br>RW | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf. |
| 3 | 0x0<br>RW | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction. |
| 2 | 0x0<br>RW | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf. |
| 1 | 0x0<br>RW | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction. |
| 0 | 0x0<br>RW | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf. |

## 44.3.6 First Uncorrectable Error Register (A_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6594h

First Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6594h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO/V | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO/V | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO/V | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO/V | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO/V | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO/V | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19 | 0x0<br>RO/V | **downstream data buffer parity error (downstream_data_buffer_par_err):** downstream data buffer parity error detected<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO/V | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** upstream cmd buffer parity error detected<br><br>**Power Well:** prst |
| 17 | 0x0<br>RO/V | **iosf upstream data parity error (iosf_upstream_data_par_err):** iosf upstream data parity error detected<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO/V | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** iosf upstream cmd parity error detected<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO/V | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO/V | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large.<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO/V | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO/V | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO/V | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction.<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RO/V | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf.<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction.<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf.<br><br>**Power Well:** prst |
| 1 | 0x0<br>RO/V | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RO/V | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf.<br><br>**Power Well:** prst |

## 44.3.7 Next Uncorrectable Error Register (A_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6598h

Next Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6598h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO/V | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO/V | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO/V | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO/V | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO/V | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO/V | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19 | 0x0<br>RO/V | **downstream data buffer parity error (downstream_data_buffer_par_err):** Downstream data buffer parity error detected<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO/V | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** Upstream cmd buffer parity error detected<br><br>**Power Well:** prst |
| 17 | 0x0<br>RO/V | **iosf upstream data parity error (iosf_upstream_data_par_err):** IOSF upstream data parity error detected<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO/V | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** IOSF upstream cmd parity error detected<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO/V | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO/V | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO/V | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO/V | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO/V | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction.<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RO/V | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from IOSF.<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction.<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf.<br><br>**Power Well:** prst |
| 1 | 0x0<br>RO/V | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction.<br><br>**Power Well:** prst |
| 0 | 0x0<br>RO/V | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf.<br><br>**Power Well:** prst |

## 44.3.8 Uncorrectable Error Select Register (A_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 659Ch

Selects which errors cause the UNCERRCNT register to increment. 0-do not count 1-count.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 659Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved** |
| 30 | 0x0<br>RO | **Reserved** |
| 29 | 0x0<br>RO | **Reserved** |
| 28 | 0x0<br>RO | **Reserved** |
| 27 | 0x0<br>RO | **Reserved** |
| 26 | 0x0<br>RO | **Reserved** |
| 25 | 0x0<br>RW | **iommu data iotlb parity error (iommu_data_iotlb_par_err):** IOMMU IOTLB data parity error |
| 24 | 0x0<br>RW | **iommu data cc parity error (iommu_data_cc_par_err):** IOMMU context cache data parity error |
| 23 | 0x0<br>RW | **iommu data l4 parity error (iommu_data_l4_par_err):** IOMMU level 4 page walk data parity error |
| 22 | 0x0<br>RW | **iommu data l3 parity error (iommu_data_l3_par_err):** IOMMU level 3 page walk data parity error |
| 21 | 0x0<br>RW | **iommu data l2 parity error (iommu_data_l2_par_err):** IOMMU level 2 page walk data parity error |
| 20 | 0x0<br>RW | **downstream cmd buffer parity error (downstream_cmd_buffer_par_err):** downstream cmd buffer parity error detected |
| 19 | 0x0<br>RW | **downstream data buffer parity error (downstream_data_buffer_par_err):** downstream data buffer parity error detected |
| 18 | 0x0<br>RW | **upstream cmd buffer parity error (upstream_cmd_buffer_par_err):** upstream cmd buffer parity error detected |
| 17 | 0x0<br>RW | **iosf upstream data parity error (iosf_upstream_data_par_err):** iosf upstream data parity error detected |
| 16 | 0x0<br>RW | **iosf upstream cmd parity error (iosf_upstream_cmd_par_err):** iosf upstream cmd parity error detected |
| 15 | 0x0<br>RO | **Reserved** |
| 14 | 0x0<br>RO | **Reserved** |
| 13 | 0x0<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 12 | 0x0<br>RO | **Reserved** |
| 11 | 0x0<br>RO | **Reserved** |
| 10 | 0x0<br>RW | **msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set |
| 9 | 0x0<br>RW | **gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large |
| 8 | 0x0<br>RW | **illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction |
| 7 | 0x0<br>RW | **at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value |
| 6 | 0x0<br>RO | **Reserved** |
| 5 | 0x0<br>RW | **bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction. |
| 4 | 0x0<br>RW | **received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf. |
| 3 | 0x0<br>RW | **bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction. |
| 2 | 0x0<br>RW | **illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf. |
| 1 | 0x0<br>RW | **bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction. |
| 0 | 0x0<br>RW | **illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf. |

## 44.3.9 Uncorrectable Error Count (A_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 65A0h

Counts the number of uncorrectable errors that have occurred.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65A0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/1C | **Overflow (OVERFLOW):** Error count overflow.<br><br>**Power Well:** prst |
| 14:0 | 0x0<br>RW/1C | **Error Count (ERR_CNT):** Uncorrectable error count.<br><br>**Power Well:** prst |

## 44.3.10 Correctable Error Status Register (A_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 65A4h

Errors that have been seen in A-Unit. This will always log errors irrespective of the CORERRMSK bit setting

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65A4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 4 | 0x0<br>RW/1C | **iommu tag iotlb parity error (iommu_tag_iotlb_par_err):** IOMMU iotlb tag parity error detected<br><br>**Power Well:** prst |
| 3 | 0x0<br>RW/1C | **iommu tag cc parity error (iommu_tag_cc_par_err):** IOMMU context cache tag parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C | **iommu tag l4 parity error (iommu_tag_l4_par_err):** IOMMU l4 cache tag parity error detected<br><br>**Power Well:** prst |
| 1 | 0x0<br>RW/1C | **iommu tag l3 parity error (iommu_tag_l3_par_err):** IOMMU l3 cache tag parity error detected<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW/1C | **iommu tag l2 parity error (iommu_tag_l2_par_err):** IOMMU l2 cache tag parity error detected<br><br>**Power Well:** prst |

## 44.3.11 Correctable Error Mask Register (A_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 65A8h

Masks where a reported error is signaled to the IEH. 1- do not signal. 0 - signal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65A8h

**Default:** FFFFFFFFh

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Reserved** |
| 30 | 0x1<br>RO | **Reserved** |
| 29 | 0x1<br>RO | **Reserved** |
| 28 | 0x1<br>RO | **Reserved** |
| 27 | 0x1<br>RO | **Reserved** |
| 26 | 0x1<br>RO | **Reserved** |
| 25 | 0x1<br>RO | **Reserved** |
| 24 | 0x1<br>RO | **Reserved** |
| 23 | 0x1<br>RO | **Reserved** |
| 22 | 0x1<br>RO | **Reserved** |
| 21 | 0x1<br>RO | **Reserved** |
| 20 | 0x1<br>RO | **Reserved** |
| 19 | 0x1<br>RO | **Reserved** |
| 18 | 0x1<br>RO | **Reserved** |
| 17 | 0x1<br>RO | **Reserved** |
| 16 | 0x1<br>RO | **Reserved** |
| 15 | 0x1<br>RO | **Reserved** |
| 14 | 0x1<br>RO | **Reserved** |
| 13 | 0x1<br>RO | **Reserved** |
| 12 | 0x1<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x1<br>RO | **Reserved** |
| 10 | 0x1<br>RO | **Reserved** |
| 9 | 0x1<br>RO | **Reserved** |
| 8 | 0x1<br>RO | **Reserved** |
| 7 | 0x1<br>RO | **Reserved** |
| 6 | 0x1<br>RO | **Reserved** |
| 5 | 0x1<br>RO | **Reserved** |
| 4 | 0x1<br>RW | **iommu tag iotlb parity error (iommu_tag_iotlb_par_err):** IOMMU iotlb tag parity error detected |
| 3 | 0x1<br>RW | **iommu tag cc parity error (iommu_tag_cc_par_err):** IOMMU context cache tag parity error detected |
| 2 | 0x1<br>RW | **iommu tag l4 parity error (iommu_tag_l4_par_err):** IOMMU l4 cache tag parity error detected |
| 1 | 0x1<br>RW | **iommu tag l3 parity error (iommu_tag_l3_par_err):** IOMMU l3 cache tag parity error detected |
| 0 | 0x1<br>RW | **iommu tag l2 parity error (iommu_tag_l2_par_err):** IOMMU l2 cache tag parity error detected |

## 44.3.12 First Correctable Error Register (A_CR_CORFERRSTS_0_0_0_MCHBAR)—Offset 65ACh

First Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65ACh

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 4 | 0x0<br>RO/V | **iommu tag iotlb parity error (iommu_tag_iotlb_par_err):** IOMMU iotlb tag parity error detected<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **iommu tag cc parity error (iommu_tag_cc_par_err):** IOMMU context cache tag parity error detected<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **iommu tag l4 parity error (iommu_tag_l4_par_err):** IOMMU l4 cache tag parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO/V | **iommu tag l3 parity error (iommu_tag_l3_par_err):** IOMMU l3 cache tag parity error detected<br><br>**Power Well:** prst |
| 0 | 0x0<br>RO/V | **iommu tag l2 parity error (iommu_tag_l2_par_err):** IOMMU l2 cache tag parity error detected<br><br>**Power Well:** prst |

## 44.3.13 Next Correctable Error Register (A_CR_CORNERRSTS_0_0_0_MCHBAR)—Offset 65B0h

Next Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65B0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 4 | 0x0<br>RO/V | **iommu tag iotlb parity error (iommu_tag_iotlb_par_err):** IOMMU iotlb tag parity error detected<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **iommu tag cc parity error (iommu_tag_cc_par_err):** IOMMU context cache tag parity error detected<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **iommu tag l4 parity error (iommu_tag_l4_par_err):** IOMMU l4 cache tag parity error detected<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO/V | **iommu tag l3 parity error (iommu_tag_l3_par_err):** IOMMU l3 cache tag parity error detected<br><br>**Power Well:** prst |
| 0 | 0x0<br>RO/V | **iommu tag l2 parity error (iommu_tag_l2_par_err):** IOMMU l2 cache tag parity error detected<br><br>**Power Well:** prst |

## 44.3.14 Correctable Error Count Select (A_CR_CORERRCNTSEL_0_0_0_MCHBAR)—Offset 65B4h

Selects which errors cause the CORERRCNT register to increment. 0-do not count 1-count.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FFFFFFFFh

**BAR and Offset:** [MCHBAR] + 65B4h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Reserved** |
| 30 | 0x1<br>RO | **Reserved** |
| 29 | 0x1<br>RO | **Reserved** |
| 28 | 0x1<br>RO | **Reserved** |
| 27 | 0x1<br>RO | **Reserved** |
| 26 | 0x1<br>RO | **Reserved** |
| 25 | 0x1<br>RO | **Reserved** |
| 24 | 0x1<br>RO | **Reserved** |
| 23 | 0x1<br>RO | **Reserved** |
| 22 | 0x1<br>RO | **Reserved** |
| 21 | 0x1<br>RO | **Reserved** |
| 20 | 0x1<br>RO | **Reserved** |
| 19 | 0x1<br>RO | **Reserved** |
| 18 | 0x1<br>RO | **Reserved** |
| 17 | 0x1<br>RO | **Reserved** |
| 16 | 0x1<br>RO | **Reserved** |
| 15 | 0x1<br>RO | **Reserved** |
| 14 | 0x1<br>RO | **Reserved** |
| 13 | 0x1<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 12 | 0x1<br>RO | **Reserved** |
| 11 | 0x1<br>RO | **Reserved** |
| 10 | 0x1<br>RO | **Reserved** |
| 9 | 0x1<br>RO | **Reserved** |
| 8 | 0x1<br>RO | **Reserved** |
| 7 | 0x1<br>RO | **Reserved** |
| 6 | 0x1<br>RO | **Reserved** |
| 5 | 0x1<br>RO | **Reserved** |
| 4 | 0x1<br>RW | **iommu tag iotlb parity error (iommu_tag_iotlb_par_err):** IOMMU iotlb tag parity error detected |
| 3 | 0x1<br>RW | **iommu tag cc parity error (iommu_tag_cc_par_err):** IOMMU context cache tag parity error detected |
| 2 | 0x1<br>RW | **iommu tag l4 parity error (iommu_tag_l4_par_err):** IOMMU l4 cache tag parity error detected |
| 1 | 0x1<br>RW | **iommu tag l3 parity error (iommu_tag_l3_par_err):** IOMMU l3 cache tag parity error detected |
| 0 | 0x1<br>RW | **iommu tag l2 parity error (iommu_tag_l2_par_err):** IOMMU l2 cache tag parity error detected |

## 44.3.15 Correctable Error Count (A_CR_CORERRCNT_0_0_0_MCHBAR)—Offset 65B8h

Counts the number of correctable errors that have occurred

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65B8h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/1C | **Overflow (OVERFLOW):** Error count overflowed.<br><br>**Power Well:** prst |
| 14:0 | 0x0<br>RW/1C | **Error Count (ERR_CNT):** Correctable error count<br><br>**Power Well:** prst |

## 44.3.16 Slice and Channel Hash (A_CR_SLICE_CHANNEL_HASH_0_0_0_MCHBAR)—Offset 65C0h

A-Unit slice and channel hash function.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 65C0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **Lock (LOCK):** Intended usage is for BIOS to set the LOCK when it updates the CR. |
| 62:52 | 0x0<br>RO | **RSVD0:** Reserved |
| 51:38 | 0x0<br>RW | **Channel Hash Mask (CH_HASH_MASK):** When both PMI channels in a slice are enabled, this field specifies the Channel Hash Mask to be applied on Addr[19:6] postremap DRAM address of the request to compute which PMI channel a request must be routed to.<br>Relevant only when HVM mode is disabled and only for requests that do not fall under the MOT region.<br>B-Unit will override the programmed value to include the Channel Selector bit See SLICEHASH.INTERLEAVE_MODE field.<br>Note that HVM mode and MOT regions have special hash requirements and hence they do not use the CH_HASH_MASK. |
| 37:33 | 0x0<br>RO | **RSVD1:** Reserved |
| 32 | 0x0<br>RW | **Channel 1 Disabled (CH_1_DISABLED):** Channel 1 in both slices are disabled no memory address mapped to ch 1. All requests sent to channel 0. |
| 31 | 0x0<br>RW | **Spare (SPARE):** No identified usage yet. |
| 30:20 | 0x0<br>RO | **RSVD2:** Reserved |
| 19:6 | 0x0<br>RW | **Slice Hash Mask (SLICE_HASH_MASK):** When both slices are enabled this field specifies the Slice Hash Mask to be applied on Addr[19:6] physical address of the request to compute which slice a request must be routed to.<br>Relevant only when HVM mode is disabled and only for physical addresses that do not fall under the Asymmetric Memory Region and the MOT region.<br>B-Unit will override the programmed value to include the Slice Selector bit See INTERLEAVE_MODE field.<br>Note that HVM mode non-address IDI requests asymmetric memory region and MOT regions have special hash requirements and hence they do not use the SLICE_HASH_MASK. |
| 5 | 0x0<br>RO | **RSVD3:** Reserved |
| 4 | 0x0<br>RW | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:2 | 0x0<br>RW | **Interleave Mode (INTERLEAVE_MODE):** Default interleave mode that specifies how the Slice Selector and Channel Selector bits are to be determined. Relevant only when HVM mode is disabled and only for system memory addresses that do not fall under the MOT region or the Asymmetric memory region in the System Address Map. Legal encodings are 0x0 0x1 and 0x2. An encoding of 0x3 is treated as if it was 0x2. When both slices and all four PMI channels are enabled:<br>• 0h: Default Slice Selector is Addr[10] and Default Channel Selector is Addr[11]<br>• 1h: Default Slice Selector is Addr[11] and Default Channel Selector is Addr[12]<br>• 2h: Default Slice Selector is Addr[12] and Default Channel Selector is Addr[13]<br>When both slices are enabled but only one channel in each slice enabled:<br>• 0h: Default Slice Selector is Addr[10]<br>• 1h: Default Slice Selector is Addr[11]<br>• 2h: Default Slice Selector is Addr[12]<br>When only SLICE0 is enabled and both channels on that slice are enabled:<br>• 0h: Default Channel Selector is Addr[10]<br>• 1h: Default Channel Selector is Addr[11]<br>• 2h: Default Channel Selector is Addr[12]<br>When SLICE0 and only one channel in that slice is enabled this field is not relevant. B-Unit overrides the setting of the SLICE_HASH_MASK to always include the Slice Selector bit. Similarly, B-Unit overrides the setting of the CH_HASH_MASK to always include the Channel Selector bit. |
| 1 | 0x0<br>RW | **Reserved** |
| 0 | 0x0<br>RW | **Slice 1 Disabled (SLICE_1_DISABLED):** Slice 1 is disabled; no memory address mapped to Slice 1. All request sent to Slice 0. |

## 44.3.17 A_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR—Offset 65D0h

Specification of asymmetric memory region 0 (in slice 0) for the configuration with 2 asymmetric memory regions.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65D0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **SLICE0_ASYM_ENABLE:** Setting this bit to 0 disables asymmetric memory region 0; setting it to 1 enables the region. |
| 30:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:20 | 0x0<br>RW | **SLICE0_ASYM_LIMIT:** Specifies bits [38:31] of the highest address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's highest address are equal to 1. |
| 19:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11:4 | 0x0<br>RW | **SLICE0_ASYM_BASE:** Specifies bits [38:31] of the base address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's base address are equal to 0. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 44.3.18 A_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR—Offset 65D4h

Specification of asymmetric memory region 1 (in slice 1) for the configuration with 2 asymmetric memory regions.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 65D4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **SLICE1_ASYM_ENABLE:** Setting this bit to 0 disables asymmetric memory region 1; setting it to 1 enables the region. |
| 30:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:20 | 0x0<br>RW | **SLICE1_ASYM_LIMIT:** Specifies bits [38:31] of the highest address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's highest address are equal to 1. |
| 19:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11:4 | 0x0<br>RW | **SLICE1_ASYM_BASE:** Specifies bits [38:31] of the base address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's base address are equal to 0. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 44.3.19 B-Unit Miscellaneous Configuration (B_CR_BMISC_0_0_0_MCHBAR)—Offset 6800h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000007h

**BAR and Offset:** [MCHBAR] + 6800h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 3 | 0x0<br>RW | **Send Boot Vector to DRAM (SEND_BOOT_VECTOR_TO_DRAM):** When set, IA accesses to 0xFFFF_0000 to 0XFFFF_FFFF will be sent to memory, regardless of the Host IO Boundary setting in TOLUD. |
| 2 | 0x1<br>RW | **ABSeginDRAM Legacy Video Area (ABSEGINDRAM):** When this bit is set, reads and writes targeting A or Bsegments are routed to DRAM. Asegment corresponds to the memory range 0xA_0000 to 0xA_FFFF. Bsegment corresponds to the memory ranges 0xB_0000 to 0xB_7FFF and 0xB_8000 to 0xB_FFFF. |
| 1 | 0x1<br>RW | **Read FSeg from DRAM (READ_FSEG_FROM_DRAM):** Writes and reads targeting Fsegment are routed to DRAM. Fsegment corresponds to the memory range 0xF_0000 to 0xF_FFFF.<br>This bit must be set '1'. |
| 0 | 0x1<br>RW | **Read ESeg from DRAM (READ_ESEG_FROM_DRAM):** Writes and reads targeting Esegment are routed to DRAM. Esegment corresponds to the memory range 0xE_0000 to 0xE_FFFF.<br>This bit must be set '1'. |

## 44.3.20 Slice 0 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE0)—Offset 6868h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6868h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Memory Access Count (MEM_ACCESS_COUNT):** Counts the number of PMI transactions that the B-Unit has sent to any PMI channel. Counts both reads and writes. Counter is not saturating and will roll over to zero. It is up to the consumer of the counter to handle roll over cases. |

## 44.3.21 Slice 1 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE1)—Offset 686Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 686Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Memory Access Count (MEM_ACCESS_COUNT):** Counts the number of PMI transactions that the B-Unit has sent to any PMI channel. Counts both reads and writes. Counter is not saturating and will roll over to zero. It is up to the consumer of the counter to handle roll over cases. |

## 44.3.22 IMR0 Base (B_CR_BIMR0BASE_0_0_0_MCHBAR)—Offset 6870h

This register, along with IMR0MASK, IMR0RAC, and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR0RAC and IMR0WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6870h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 28:0 | 0x0<br>RW | **Base 0 IMR0 Base (IMR0_BASE):** Specifies bits 38:10 of the start address of IMR0 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR0MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR0 defined region. |

## 44.3.23 IMR0 Mask (B_CR_BIMR0MASK_0_0_0_MCHBAR)—Offset 6874h

This register, along with IMR0BASE, IMR0RAC, and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR0RAC and IMR0WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6874h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 28:0 | 0x0<br>RW | **Mask 0 IMR0 Mask (IMR0_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR0BASE[28:0] value. A match indicates that the incoming address falls within the IMR0 region. |

## 44.3.24 IMR0 Control Policy (B_CR_BIMR0CP_0_0_0_MCHBAR)—Offset 6878h

This register controls the access policy to the Read Access Policy BIMR0RAC, Write Access Policy BIMR0WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6878h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR0 Control Policy (IMR0_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field. |

## 44.3.25 IMR0 Read Access Policy (B_CR_BIMR0RAC_0_0_0_MCHBAR)—Offset 6880h

This register, along with IMR0BASE, IMR0MASK and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR0. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6880h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR0 Read Access Policy 63 (IMR0_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR0 Read Access Policy 62 (IMR0_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR0 Read Access Policy 61 (IMR0_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR0 Read Access Policy 60 (IMR0_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR0 Read Access Policy 59 (IMR0_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR0 Read Access Policy 58 (IMR0_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR0 Read Access Policy 57 (IMR0_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR0 Read Access Policy 56 (IMR0_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR0 Read Access Policy 55 (IMR0_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR0 Read Access Policy 54 (IMR0_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR0 Read Access Policy 53 (IMR0_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR0 Read Access Policy 52 (IMR0_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR0 Read Access Policy 51 (IMR0_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR0 Read Access Policy 50 (IMR0_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR0 Read Access Policy 49 (IMR0_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR0 Read Access Policy 48 (IMR0_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR0 Read Access Policy 47 (IMR0_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR0 Read Access Policy 46 (IMR0_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR0 Read Access Policy 45 (IMR0_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR0 Read Access Policy 44 (IMR0_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR0 Read Access Policy 43 (IMR0_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR0 Read Access Policy 42 (IMR0_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR0 Read Access Policy 41 (IMR0_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR0 Read Access Policy 40 (IMR0_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR0 Read Access Policy 39 (IMR0_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR0 Read Access Policy 38 (IMR0_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR0 Read Access Policy 37 (IMR0_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR0 Read Access Policy 36 (IMR0_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR0 Read Access Policy 35 (IMR0_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR0 Read Access Policy 34 (IMR0_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR0 Read Access Policy 33 (IMR0_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR0 Read Access Policy 32 (IMR0_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR0 Read Access Policy 31 (IMR0_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR0 Read Access Policy 30 (IMR0_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR0 Read Access Policy 29 (IMR0_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR0 Read Access Policy 28 (IMR0_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR0 Read Access Policy 27 (IMR0_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR0 Read Access Policy 26 (IMR0_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR0 Read Access Policy 25 (IMR0_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR0 Read Access Policy 24 (IMR0_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR0 Read Access Policy 23 (IMR0_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR0 Read Access Policy 22 (IMR0_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR0 Read Access Policy 21 (IMR0_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR0 Read Access Policy 20 (IMR0_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR0 Read Access Policy 19 (IMR0_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR0 Read Access Policy 18 (IMR0_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR0 Read Access Policy 17 (IMR0_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR0 Read Access Policy 16 (IMR0_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR0 Read Access Policy 15 (IMR0_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR0 Read Access Policy 14 (IMR0_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR0 Read Access Policy 13 (IMR0_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR0 Read Access Policy 12 (IMR0_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR0 Read Access Policy 11 (IMR0_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR0 Read Access Policy 10 (IMR0_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR0 Read Access Policy 9 (IMR0_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR0 Read Access Policy 8 (IMR0_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR0 Read Access Policy 7 (IMR0_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR0 Read Access Policy 6 (IMR0_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR0 Read Access Policy 5 (IMR0_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR0 Read Access Policy 4 (IMR0_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR0 Read Access Policy 3 (IMR0_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR0 Read Access Policy 2 (IMR0_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR0 Read Access Policy 1 (IMR0_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR0 Read Access Policy 0 (IMR0_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

## 44.3.26 IMR0 Write Access Policy (B_CR_BIMR0WAC_0_0_0_MCHBAR)—Offset 6888h

This register, along with IMR0BASE, IMR0MASK and IMR0RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR0. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6888h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR0 Write Access Policy 63 (IMR0_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR0 Write Access Policy 62 (IMR0_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR0 Write Access Policy 61 (IMR0_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR0 Write Access Policy 60 (IMR0_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR0 Write Access Policy 59 (IMR0_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR0 Write Access Policy 58 (IMR0_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR0 Write Access Policy 57 (IMR0_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR0 Write Access Policy 56 (IMR0_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR0 Write Access Policy 55 (IMR0_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR0 Write Access Policy 54 (IMR0_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR0 Write Access Policy 53 (IMR0_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR0 Write Access Policy 52 (IMR0_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR0 Write Access Policy 51 (IMR0_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR0 Write Access Policy 50 (IMR0_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR0 Write Access Policy 49 (IMR0_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR0 Write Access Policy 48 (IMR0_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR0 Write Access Policy 47 (IMR0_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR0 Write Access Policy 46 (IMR0_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR0 Write Access Policy 45 (IMR0_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR0 Write Access Policy 44 (IMR0_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR0 Write Access Policy 43 (IMR0_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR0 Write Access Policy 42 (IMR0_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR0 Write Access Policy 41 (IMR0_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR0 Write Access Policy 40 (IMR0_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR0 Write Access Policy 39 (IMR0_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR0 Write Access Policy 38 (IMR0_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR0 Write Access Policy 37 (IMR0_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR0 Write Access Policy 36 (IMR0_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR0 Write Access Policy 35 (IMR0_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR0 Write Access Policy 34 (IMR0_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR0 Write Access Policy 33 (IMR0_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR0 Write Access Policy 32 (IMR0_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR0 Write Access Policy 31 (IMR0_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR0 Write Access Policy 30 (IMR0_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR0 Write Access Policy 29 (IMR0_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR0 Write Access Policy 28 (IMR0_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR0 Write Access Policy 27 (IMR0_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR0 Write Access Policy 26 (IMR0_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR0 Write Access Policy 25 (IMR0_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR0 Write Access Policy 24 (IMR0_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR0 Write Access Policy 23 (IMR0_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR0 Write Access Policy 22 (IMR0_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR0 Write Access Policy 21 (IMR0_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR0 Write Access Policy 20 (IMR0_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR0 Write Access Policy 19 (IMR0_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR0 Write Access Policy 18 (IMR0_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR0 Write Access Policy 17 (IMR0_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR0 Write Access Policy 16 (IMR0_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR0 Write Access Policy 15 (IMR0_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR0 Write Access Policy 14 (IMR0_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR0 Write Access Policy 13 (IMR0_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR0 Write Access Policy 12 (IMR0_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR0 Write Access Policy 11 (IMR0_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR0 Write Access Policy 10 (IMR0_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR0 Write Access Policy 9 (IMR0_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR0 Write Access Policy 8 (IMR0_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR0 Write Access Policy 7 (IMR0_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR0 Write Access Policy 6 (IMR0_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR0 Write Access Policy 5 (IMR0_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR0 Write Access Policy 4 (IMR0_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR0 Write Access Policy 3 (IMR0_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR0 Write Access Policy 2 (IMR0_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR0 Write Access Policy 1 (IMR0_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR0 Write Access Policy 0 (IMR0_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value. |

## 44.3.27 B_CR_BIMR1BASE_0_0_0_MCHBAR—Offset 6890h

This register, along with IMR1MASK, IMR1RAC and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR1RAC and IMR1WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6890h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Trace Enable (TR_EN):** Asset Classification AC[0]: Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 1 IMR1 Base (IMR1_BASE):** Specifies bits 38:10 of the start address of IMR1 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR1MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR1 defined region. |

## 44.3.28 IMR1 Mask (B_CR_BIMR1MASK_0_0_0_MCHBAR)—Offset 6894h

This register, along with IMR1BASE, IMR1RAC and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR1RAC and IMR1WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6894h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GT Implicit Writeback Enable (GT_IWB_EN):** Asset Classification AC[2]: Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requestor of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 1 IMR1 Mask (IMR1_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR1BASE[28:0] value. A match indicates that the incoming address falls within the IMR1 region. |

## 44.3.29 IMR1 Control Policy (B_CR_BIMR1CP_0_0_0_MCHBAR)—Offset 6898h

This register controls the access policy to the Read Access Policy BIMR1RAC, the Write Access Policy BIMR1WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6898h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR1 Control Policy (IMR1_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.30 IMR1 Read Access Policy (B_CR_BIMR1RAC_0_0_0_MCHBAR)—Offset 68A0h

This register, along with IMR1BASE, IMR1MASK and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68A0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR1 Read Access Policy 63 (IMR1_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR1 Read Access Policy 62 (IMR1_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR1 Read Access Policy 61 (IMR1_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR1 Read Access Policy 60 (IMR1_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR1 Read Access Policy 59 (IMR1_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR1 Read Access Policy 58 (IMR1_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR1 Read Access Policy 57 (IMR1_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR1 Read Access Policy 56 (IMR1_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR1 Read Access Policy 55 (IMR1_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR1 Read Access Policy 54 (IMR1_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR1 Read Access Policy 53 (IMR1_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR1 Read Access Policy 52 (IMR1_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR1 Read Access Policy 51 (IMR1_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR1 Read Access Policy 50 (IMR1_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR1 Read Access Policy 49 (IMR1_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR1 Read Access Policy 48 (IMR1_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR1 Read Access Policy 47 (IMR1_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR1 Read Access Policy 46 (IMR1_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR1 Read Access Policy 45 (IMR1_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR1 Read Access Policy 44 (IMR1_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR1 Read Access Policy 43 (IMR1_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR1 Read Access Policy 42 (IMR1_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR1 Read Access Policy 41 (IMR1_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR1 Read Access Policy 40 (IMR1_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR1 Read Access Policy 39 (IMR1_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR1 Read Access Policy 38 (IMR1_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR1 Read Access Policy 37 (IMR1_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR1 Read Access Policy 36 (IMR1_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR1 Read Access Policy 35 (IMR1_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR1 Read Access Policy 34 (IMR1_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR1 Read Access Policy 33 (IMR1_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR1 Read Access Policy 32 (IMR1_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR1 Read Access Policy 31 (IMR1_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR1 Read Access Policy 30 (IMR1_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR1 Read Access Policy 29 (IMR1_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR1 Read Access Policy 28 (IMR1_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR1 Read Access Policy 27 (IMR1_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR1 Read Access Policy 26 (IMR1_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR1 Read Access Policy 25 (IMR1_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR1 Read Access Policy 24 (IMR1_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR1 Read Access Policy 23 (IMR1_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR1 Read Access Policy 22 (IMR1_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR1 Read Access Policy 21 (IMR1_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR1 Read Access Policy 20 (IMR1_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR1 Read Access Policy 19 (IMR1_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR1 Read Access Policy 18 (IMR1_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR1 Read Access Policy 17 (IMR1_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR1 Read Access Policy 16 (IMR1_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR1 Read Access Policy 15 (IMR1_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR1 Read Access Policy 14 (IMR1_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR1 Read Access Policy 13 (IMR1_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR1 Read Access Policy 12 (IMR1_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR1 Read Access Policy 11 (IMR1_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR1 Read Access Policy 10 (IMR1_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR1 Read Access Policy 9 (IMR1_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR1 Read Access Policy 8 (IMR1_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR1 Read Access Policy 7 (IMR1_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR1 Read Access Policy 6 (IMR1_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR1 Read Access Policy 5 (IMR1_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR1 Read Access Policy 4 (IMR1_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR1 Read Access Policy 3 (IMR1_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR1 Read Access Policy 2 (IMR1_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR1 Read Access Policy 1 (IMR1_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR1 Read Access Policy 0 (IMR1_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

## 44.3.31 IMR1 Write Access Policy (B_CR_BIMR1WAC_0_0_0_MCHBAR)—Offset 68A8h

This register, along with IMR1BASE, IMR1MASK and IMR1RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68A8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR1 Write Access Policy 63 (IMR1_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR1 Write Access Policy 62 (IMR1_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR1 Write Access Policy 61 (IMR1_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR1 Write Access Policy 60 (IMR1_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR1 Write Access Policy 59 (IMR1_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR1 Write Access Policy 58 (IMR1_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR1 Write Access Policy 57 (IMR1_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR1 Write Access Policy 56 (IMR1_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR1 Write Access Policy 55 (IMR1_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR1 Write Access Policy 54 (IMR1_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR1 Write Access Policy 53 (IMR1_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR1 Write Access Policy 52 (IMR1_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR1 Write Access Policy 51 (IMR1_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR1 Write Access Policy 50 (IMR1_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR1 Write Access Policy 49 (IMR1_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR1 Write Access Policy 48 (IMR1_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR1 Write Access Policy 47 (IMR1_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR1 Write Access Policy 46 (IMR1_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR1 Write Access Policy 45 (IMR1_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR1 Write Access Policy 44 (IMR1_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR1 Write Access Policy 43 (IMR1_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR1 Write Access Policy 42 (IMR1_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR1 Write Access Policy 41 (IMR1_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR1 Write Access Policy 40 (IMR1_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR1 Write Access Policy 39 (IMR1_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR1 Write Access Policy 38 (IMR1_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR1 Write Access Policy 37 (IMR1_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR1 Write Access Policy 36 (IMR1_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR1 Write Access Policy 35 (IMR1_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR1 Write Access Policy 34 (IMR1_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR1 Write Access Policy 33 (IMR1_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR1 Write Access Policy 32 (IMR1_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR1 Write Access Policy 31 (IMR1_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR1 Write Access Policy 30 (IMR1_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR1 Write Access Policy 29 (IMR1_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR1 Write Access Policy 28 (IMR1_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR1 Write Access Policy 27 (IMR1_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR1 Write Access Policy 26 (IMR1_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR1 Write Access Policy 25 (IMR1_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR1 Write Access Policy 24 (IMR1_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR1 Write Access Policy 23 (IMR1_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR1 Write Access Policy 22 (IMR1_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR1 Write Access Policy 21 (IMR1_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR1 Write Access Policy 20 (IMR1_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR1 Write Access Policy 19 (IMR1_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR1 Write Access Policy 18 (IMR1_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR1 Write Access Policy 17 (IMR1_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR1 Write Access Policy 16 (IMR1_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR1 Write Access Policy 15 (IMR1_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR1 Write Access Policy 14 (IMR1_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR1 Write Access Policy 13 (IMR1_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR1 Write Access Policy 12 (IMR1_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR1 Write Access Policy 11 (IMR1_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR1 Write Access Policy 10 (IMR1_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR1 Write Access Policy 9 (IMR1_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR1 Write Access Policy 8 (IMR1_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR1 Write Access Policy 7 (IMR1_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR1 Write Access Policy 6 (IMR1_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR1 Write Access Policy 5 (IMR1_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR1 Write Access Policy 4 (IMR1_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR1 Write Access Policy 3 (IMR1_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR1 Write Access Policy 2 (IMR1_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR1 Write Access Policy 1 (IMR1_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR1 Write Access Policy 0 (IMR1_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value. |

## 44.3.32 Base 0 IMR2 Base (B_CR_BIMR2BASE_0_0_0_MCHBAR)—Offset 68B0h

This register, along with IMR2MASK, IMR2RAC and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR2RAC and IMR2WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68B0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR2 Base (IMR2_BASE):** Specifies bits 38:10 of the start address of IMR2 region. IMR region size must be a strict poweroftwo, at least 1KB and naturally aligned to the size. These bits are compared with the result of the IMR2MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR2 defined region. |

## 44.3.33 IMR2 Mask (B_CR_BIMR2MASK_0_0_0_MCHBAR)—Offset 68B4h

This register, along with IMR2BASE, IMR2RAC, and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR2RAC and IMR2WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68B4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestpr depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR2 Mask (IMR2_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR2BASE[28:0] value. A match indicates that the incoming address falls within the IMR2 region. |

## 44.3.34 IMR2 Control Policy (B_CR_BIMR2CP_0_0_0_MCHBAR)—Offset 68B8h

This register controls the access policy to the Read Access Policy BIMR2RAC, Write Access Policy BIMR2WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68B8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR2 Control Policy (IMR2_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.35 IMR2 Read Access Policy (B_CR_BIMR2RAC_0_0_0_MCHBAR)—Offset 68C0h

This register, along with IMR2BASE, IMR2MASK and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR2. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68C0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR2 Read Access Policy 63 (IMR2_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR2 Read Access Policy 62 (IMR2_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR2 Read Access Policy 61 (IMR2_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR2 Read Access Policy 60 (IMR2_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR2 Read Access Policy 59 (IMR2_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR2 Read Access Policy 58 (IMR2_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR2 Read Access Policy 57 (IMR2_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR2 Read Access Policy 56 (IMR2_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR2 Read Access Policy 55 (IMR2_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR2 Read Access Policy 54 (IMR2_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR2 Read Access Policy 53 (IMR2_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR2 Read Access Policy 52 (IMR2_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR2 Read Access Policy 51 (IMR2_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR2 Read Access Policy 50 (IMR2_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR2 Read Access Policy 49 (IMR2_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR2 Read Access Policy 48 (IMR2_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR2 Read Access Policy 47 (IMR2_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR2 Read Access Policy 46 (IMR2_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR2 Read Access Policy 45 (IMR2_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR2 Read Access Policy 44 (IMR2_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR2 Read Access Policy 43 (IMR2_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR2 Read Access Policy 42 (IMR2_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR2 Read Access Policy 41 (IMR2_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR2 Read Access Policy 40 (IMR2_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR2 Read Access Policy 39 (IMR2_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR2 Read Access Policy 38 (IMR2_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR2 Read Access Policy 37 (IMR2_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR2 Read Access Policy 36 (IMR2_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR2 Read Access Policy 35 (IMR2_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR2 Read Access Policy 34 (IMR2_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR2 Read Access Policy 33 (IMR2_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR2 Read Access Policy 32 (IMR2_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR2 Read Access Policy 31 (IMR2_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR2 Read Access Policy 30 (IMR2_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR2 Read Access Policy 29 (IMR2_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR2 Read Access Policy 28 (IMR2_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR2 Read Access Policy 27 (IMR2_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR2 Read Access Policy 26 (IMR2_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR2 Read Access Policy 25 (IMR2_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR2 Read Access Policy 24 (IMR2_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR2 Read Access Policy 23 (IMR2_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR2 Read Access Policy 22 (IMR2_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR2 Read Access Policy 21 (IMR2_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR2 Read Access Policy 20 (IMR2_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR2 Read Access Policy 19 (IMR2_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR2 Read Access Policy 18 (IMR2_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR2 Read Access Policy 17 (IMR2_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR2 Read Access Policy 16 (IMR2_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR2 Read Access Policy 15 (IMR2_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR2 Read Access Policy 14 (IMR2_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR2 Read Access Policy 13 (IMR2_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR2 Read Access Policy 12 (IMR2_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR2 Read Access Policy 11 (IMR2_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR2 Read Access Policy 10 (IMR2_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR2 Read Access Policy 9 (IMR2_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR2 Read Access Policy 8 (IMR2_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR2 Read Access Policy 7 (IMR2_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR2 Read Access Policy 6 (IMR2_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR2 Read Access Policy 5 (IMR2_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR2 Read Access Policy 4 (IMR2_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR2 Read Access Policy 3 (IMR2_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR2 Read Access Policy 2 (IMR2_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR2 Read Access Policy 1 (IMR2_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR2 Read Access Policy 0 (IMR2_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

## 44.3.36 IMR2 Write Access Policy (B_CR_BIMR2WAC_0_0_0_MCHBAR)—Offset 68C8h

This register, along with IMR2BASE, IMR2MASK and IMR2RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR2. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68C8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR2 Write Access Policy 63 (IMR2_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR2 Write Access Policy 62 (IMR2_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR2 Write Access Policy 61 (IMR2_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR2 Write Access Policy 60 (IMR2_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR2 Write Access Policy 59 (IMR2_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR2 Write Access Policy 58 (IMR2_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR2 Write Access Policy 57 (IMR2_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR2 Write Access Policy 56 (IMR2_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR2 Write Access Policy 55 (IMR2_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR2 Write Access Policy 54 (IMR2_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR2 Write Access Policy 53 (IMR2_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR2 Write Access Policy 52 (IMR2_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR2 Write Access Policy 51 (IMR2_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR2 Write Access Policy 50 (IMR2_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR2 Write Access Policy 49 (IMR2_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR2 Write Access Policy 48 (IMR2_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR2 Write Access Policy 47 (IMR2_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR2 Write Access Policy 46 (IMR2_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR2 Write Access Policy 45 (IMR2_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR2 Write Access Policy 44 (IMR2_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR2 Write Access Policy 43 (IMR2_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR2 Write Access Policy 42 (IMR2_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR2 Write Access Policy 41 (IMR2_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR2 Write Access Policy 40 (IMR2_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR2 Write Access Policy 39 (IMR2_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR2 Write Access Policy 38 (IMR2_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR2 Write Access Policy 37 (IMR2_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR2 Write Access Policy 36 (IMR2_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR2 Write Access Policy 35 (IMR2_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR2 Write Access Policy 34 (IMR2_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR2 Write Access Policy 33 (IMR2_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR2 Write Access Policy 32 (IMR2_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR2 Write Access Policy 31 (IMR2_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR2 Write Access Policy 30 (IMR2_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR2 Write Access Policy 29 (IMR2_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR2 Write Access Policy 28 (IMR2_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR2 Write Access Policy 27 (IMR2_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR2 Write Access Policy 26 (IMR2_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR2 Write Access Policy 25 (IMR2_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR2 Write Access Policy 24 (IMR2_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR2 Write Access Policy 23 (IMR2_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR2 Write Access Policy 22 (IMR2_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR2 Write Access Policy 21 (IMR2_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR2 Write Access Policy 20 (IMR2_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR2 Write Access Policy 19 (IMR2_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR2 Write Access Policy 18 (IMR2_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR2 Write Access Policy 17 (IMR2_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR2 Write Access Policy 16 (IMR2_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR2 Write Access Policy 15 (IMR2_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR2 Write Access Policy 14 (IMR2_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR2 Write Access Policy 13 (IMR2_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR2 Write Access Policy 12 (IMR2_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR2 Write Access Policy 11 (IMR2_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR2 Write Access Policy 10 (IMR2_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR2 Write Access Policy 9 (IMR2_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR2 Write Access Policy 8 (IMR2_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR2 Write Access Policy 7 (IMR2_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR2 Write Access Policy 6 (IMR2_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR2 Write Access Policy 5 (IMR2_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR2 Write Access Policy 4 (IMR2_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR2 Write Access Policy 3 (IMR2_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR2 Write Access Policy 2 (IMR2_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR2 Write Access Policy 1 (IMR2_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR2 Write Access Policy 0 (IMR2_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value. |

## 44.3.37 IMR3 Base (B_CR_BIMR3BASE_0_0_0_MCHBAR)—Offset 68D0h

This register, along with IMR3MASK, IMR3RAC and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR3RAC and IMR3WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68D0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR3 Base (IMR3_BASE):** Specifies bits 38:10 of the start address of IMR3 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR3MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR3 defined region. |

## 44.3.38 IMR3 Mask (B_CR_BIMR3MASK_0_0_0_MCHBAR)—Offset 68D4h

This register, along with IMR3BASE, IMR3RAC and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR3RAC and IMR3WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68D4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR3 Mask (IMR3_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR3BASE[28:0] value. A match indicates that the incoming address falls within the IMR3 region. |

## 44.3.39 IMR3 Control Policy (B_CR_BIMR3CP_0_0_0_MCHBAR)—Offset 68D8h

This register controls the access policy to the Read Access Policy BIMR3RAC, the Write Access Policy BIMR3WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68D8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR3 Control Policy (IMR3_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.40 IMR3 Read Access Policy (B_CR_BIMR3RAC_0_0_0_MCHBAR)—Offset 68E0h

This register, along with IMR3BASE, IMR3MASK and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR3. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68E0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR3 Read Access Policy 63 (IMR3_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR3 Read Access Policy 62 (IMR3_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR3 Read Access Policy 61 (IMR3_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR3 Read Access Policy 60 (IMR3_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR3 Read Access Policy 59 (IMR3_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR3 Read Access Policy 58 (IMR3_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR3 Read Access Policy 57 (IMR3_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR3 Read Access Policy 56 (IMR3_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR3 Read Access Policy 55 (IMR3_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR3 Read Access Policy 54 (IMR3_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR3 Read Access Policy 53 (IMR3_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR3 Read Access Policy 52 (IMR3_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR3 Read Access Policy 51 (IMR3_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR3 Read Access Policy 50 (IMR3_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR3 Read Access Policy 49 (IMR3_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR3 Read Access Policy 48 (IMR3_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR3 Read Access Policy 47 (IMR3_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR3 Read Access Policy 46 (IMR3_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR3 Read Access Policy 45 (IMR3_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR3 Read Access Policy 44 (IMR3_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR3 Read Access Policy 43 (IMR3_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR3 Read Access Policy 42 (IMR3_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR3 Read Access Policy 41 (IMR3_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR3 Read Access Policy 40 (IMR3_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR3 Read Access Policy 39 (IMR3_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR3 Read Access Policy 38 (IMR3_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR3 Read Access Policy 37 (IMR3_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR3 Read Access Policy 36 (IMR3_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR3 Read Access Policy 35 (IMR3_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR3 Read Access Policy 34 (IMR3_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR3 Read Access Policy 33 (IMR3_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR3 Read Access Policy 32 (IMR3_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR3 Read Access Policy 31 (IMR3_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR3 Read Access Policy 30 (IMR3_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR3 Read Access Policy 29 (IMR3_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR3 Read Access Policy 28 (IMR3_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR3 Read Access Policy 27 (IMR3_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR3 Read Access Policy 26 (IMR3_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR3 Read Access Policy 25 (IMR3_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR3 Read Access Policy 24 (IMR3_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR3 Read Access Policy 23 (IMR3_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR3 Read Access Policy 22 (IMR3_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR3 Read Access Policy 21 (IMR3_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR3 Read Access Policy 20 (IMR3_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR3 Read Access Policy 19 (IMR3_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR3 Read Access Policy 18 (IMR3_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR3 Read Access Policy 17 (IMR3_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR3 Read Access Policy 16 (IMR3_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR3 Read Access Policy 15 (IMR3_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR3 Read Access Policy 14 (IMR3_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR3 Read Access Policy 13 (IMR3_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR3 Read Access Policy 12 (IMR3_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR3 Read Access Policy 11 (IMR3_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR3 Read Access Policy 10 (IMR3_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR3 Read Access Policy 9 (IMR3_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR3 Read Access Policy 8 (IMR3_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR3 Read Access Policy 7 (IMR3_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR3 Read Access Policy 6 (IMR3_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR3 Read Access Policy 5 (IMR3_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR3 Read Access Policy 4 (IMR3_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR3 Read Access Policy 3 (IMR3_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR3 Read Access Policy 2 (IMR3_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR3 Read Access Policy 1 (IMR3_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR3 Read Access Policy 0 (IMR3_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

## 44.3.41 IMR3 Write Access Policy (B_CR_BIMR3WAC_0_0_0_MCHBAR)—Offset 68E8h

This register, along with IMR3BASE, IMR3MASK and IMR3RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR3. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68E8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR3 Write Access Policy 63 (IMR3_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR3 Write Access Policy 62 (IMR3_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR3 Write Access Policy 61 (IMR3_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR3 Write Access Policy 60 (IMR3_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR3 Write Access Policy 59 (IMR3_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR3 Write Access Policy 58 (IMR3_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR3 Write Access Policy 57 (IMR3_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR3 Write Access Policy 56 (IMR3_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR3 Write Access Policy 55 (IMR3_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR3 Write Access Policy 54 (IMR3_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR3 Write Access Policy 53 (IMR3_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR3 Write Access Policy 52 (IMR3_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR3 Write Access Policy 51 (IMR3_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR3 Write Access Policy 50 (IMR3_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR3 Write Access Policy 49 (IMR3_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR3 Write Access Policy 48 (IMR3_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR3 Write Access Policy 47 (IMR3_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR3 Write Access Policy 46 (IMR3_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR3 Write Access Policy 45 (IMR3_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR3 Write Access Policy 44 (IMR3_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR3 Write Access Policy 43 (IMR3_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR3 Write Access Policy 42 (IMR3_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR3 Write Access Policy 41 (IMR3_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR3 Write Access Policy 40 (IMR3_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR3 Write Access Policy 39 (IMR3_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR3 Write Access Policy 38 (IMR3_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR3 Write Access Policy 37 (IMR3_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR3 Write Access Policy 36 (IMR3_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR3 Write Access Policy 35 (IMR3_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR3 Write Access Policy 34 (IMR3_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR3 Write Access Policy 33 (IMR3_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR3 Write Access Policy 32 (IMR3_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR3 Write Access Policy 31 (IMR3_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR3 Write Access Policy 30 (IMR3_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR3 Write Access Policy 29 (IMR3_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR3 Write Access Policy 28 (IMR3_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR3 Write Access Policy 27 (IMR3_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR3 Write Access Policy 26 (IMR3_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR3 Write Access Policy 25 (IMR3_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR3 Write Access Policy 24 (IMR3_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR3 Write Access Policy 23 (IMR3_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR3 Write Access Policy 22 (IMR3_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR3 Write Access Policy 21 (IMR3_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR3 Write Access Policy 20 (IMR3_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR3 Write Access Policy 19 (IMR3_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR3 Write Access Policy 18 (IMR3_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR3 Write Access Policy 17 (IMR3_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR3 Write Access Policy 16 (IMR3_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR3 Write Access Policy 15 (IMR3_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR3 Write Access Policy 14 (IMR3_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR3 Write Access Policy 13 (IMR3_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR3 Write Access Policy 12 (IMR3_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR3 Write Access Policy 11 (IMR3_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR3 Write Access Policy 10 (IMR3_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR3 Write Access Policy 9 (IMR3_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR3 Write Access Policy 8 (IMR3_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR3 Write Access Policy 7 (IMR3_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR3 Write Access Policy 6 (IMR3_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR3 Write Access Policy 5 (IMR3_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR3 Write Access Policy 4 (IMR3_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR3 Write Access Policy 3 (IMR3_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR3 Write Access Policy 2 (IMR3_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR3 Write Access Policy 1 (IMR3_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR3 Write Access Policy 0 (IMR3_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value. |

## 44.3.42 IMR4 Base (B_CR_BIMR4BASE_0_0_0_MCHBAR)—Offset 68F0h

This register, along with IMR4MASK, IMR4RAC and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR4RAC and IMR4WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68F0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR4 Base (IMR4_BASE):** Specifies bits 38:10 of the start address of IMR4 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR4MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR4 defined region. |

## 44.3.43 IMR4 Mask (B_CR_BIMR4MASK_0_0_0_MCHBAR)—Offset 68F4h

This register, along with IMR4BASE, IMR4RAC and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR4RAC and IMR4WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 68F4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR4 Mask (IMR4_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR4BASE[28:0] value. A match indicates that the incoming address falls within the IMR4 region. |

## 44.3.44 B-Unit IMR4 Control Policy (B_CR_BIMR4CP_0_0_0_MCHBAR)—Offset 68F8h

This register controls the access policy to the Read Access Policy BIMR4RAC, the Write Access Policy BIMR4WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 68F8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **B-Unit IMR4 Control Policy (IMR4_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.45 IMR4 Read Access Policy (B_CR_BIMR4RAC_0_0_0_MCHBAR)—Offset 6900h

This register, along with IMR4BASE, IMR4MASK and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR4. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6900h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR4 Read Access Policy 63 (IMR4_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR4 Read Access Policy 62 (IMR4_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR4 Read Access Policy 61 (IMR4_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR4 Read Access Policy 60 (IMR4_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR4 Read Access Policy 59 (IMR4_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR4 Read Access Policy 58 (IMR4_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR4 Read Access Policy 57 (IMR4_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR4 Read Access Policy 56 (IMR4_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR4 Read Access Policy 55 (IMR4_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR4 Read Access Policy 54 (IMR4_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR4 Read Access Policy 53 (IMR4_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR4 Read Access Policy 52 (IMR4_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR4 Read Access Policy 51 (IMR4_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR4 Read Access Policy 50 (IMR4_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR4 Read Access Policy 49 (IMR4_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR4 Read Access Policy 48 (IMR4_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR4 Read Access Policy 47 (IMR4_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR4 Read Access Policy 46 (IMR4_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR4 Read Access Policy 45 (IMR4_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR4 Read Access Policy 44 (IMR4_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR4 Read Access Policy 43 (IMR4_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR4 Read Access Policy 42 (IMR4_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR4 Read Access Policy 41 (IMR4_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR4 Read Access Policy 40 (IMR4_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR4 Read Access Policy 39 (IMR4_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR4 Read Access Policy 38 (IMR4_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR4 Read Access Policy 37 (IMR4_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR4 Read Access Policy 36 (IMR4_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR4 Read Access Policy 35 (IMR4_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR4 Read Access Policy 34 (IMR4_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR4 Read Access Policy 33 (IMR4_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR4 Read Access Policy 32 (IMR4_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR4 Read Access Policy 31 (IMR4_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR4 Read Access Policy 30 (IMR4_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR4 Read Access Policy 29 (IMR4_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR4 Read Access Policy 28 (IMR4_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR4 Read Access Policy 27 (IMR4_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR4 Read Access Policy 26 (IMR4_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR4 Read Access Policy 25 (IMR4_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR4 Read Access Policy 24 (IMR4_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR4 Read Access Policy 23 (IMR4_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR4 Read Access Policy 22 (IMR4_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR4 Read Access Policy 21 (IMR4_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR4 Read Access Policy 20 (IMR4_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR4 Read Access Policy 19 (IMR4_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR4 Read Access Policy 18 (IMR4_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR4 Read Access Policy 17 (IMR4_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR4 Read Access Policy 16 (IMR4_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR4 Read Access Policy 15 (IMR4_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR4 Read Access Policy 14 (IMR4_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR4 Read Access Policy 13 (IMR4_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR4 Read Access Policy 12 (IMR4_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR4 Read Access Policy 11 (IMR4_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR4 Read Access Policy 10 (IMR4_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR4 Read Access Policy 9 (IMR4_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR4 Read Access Policy 8 (IMR4_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR4 Read Access Policy 7 (IMR4_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR4 Read Access Policy 6 (IMR4_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR4 Read Access Policy 5 (IMR4_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR4 Read Access Policy 4 (IMR4_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR4 Read Access Policy 3 (IMR4_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR4 Read Access Policy 2 (IMR4_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR4 Read Access Policy 1 (IMR4_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR4 Read Access Policy 0 (IMR4_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

## 44.3.46 IMR4 Write Access Policy (B_CR_BIMR4WAC_0_0_0_MCHBAR)—Offset 6908h

This register, along with IMR4BASE, IMR4MASK and IMR4RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR4. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6908h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR4_WRITE_POL_63:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR4_WRITE_POL_62:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR4_WRITE_POL_61:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR4_WRITE_POL_60:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR4_WRITE_POL_59:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR4_WRITE_POL_58:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR4_WRITE_POL_57:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR4_WRITE_POL_56:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR4_WRITE_POL_55:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR4_WRITE_POL_54:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR4_WRITE_POL_53:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR4_WRITE_POL_52:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR4_WRITE_POL_51:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR4_WRITE_POL_50:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR4_WRITE_POL_49:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR4_WRITE_POL_48:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR4_WRITE_POL_47:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR4_WRITE_POL_46:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR4_WRITE_POL_45:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR4_WRITE_POL_44:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR4_WRITE_POL_43:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR4_WRITE_POL_42:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR4_WRITE_POL_41:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR4_WRITE_POL_40:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR4_WRITE_POL_39:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR4_WRITE_POL_38:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR4_WRITE_POL_37:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR4_WRITE_POL_36:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR4_WRITE_POL_35:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR4_WRITE_POL_34:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR4_WRITE_POL_33:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR4_WRITE_POL_32:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR4_WRITE_POL_31:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR4_WRITE_POL_30:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR4_WRITE_POL_29:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR4_WRITE_POL_28:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR4_WRITE_POL_27:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR4_WRITE_POL_26:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR4_WRITE_POL_25:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR4_WRITE_POL_24:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR4_WRITE_POL_23:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR4_WRITE_POL_22:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR4_WRITE_POL_21:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR4_WRITE_POL_20:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR4_WRITE_POL_19:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR4_WRITE_POL_18:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR4_WRITE_POL_17:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR4_WRITE_POL_16:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR4_WRITE_POL_15:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR4_WRITE_POL_14:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR4_WRITE_POL_13:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR4_WRITE_POL_12:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR4_WRITE_POL_11:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR4_WRITE_POL_10:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR4_WRITE_POL_9:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR4_WRITE_POL_8:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR4_WRITE_POL_7:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR4_WRITE_POL_6:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR4_WRITE_POL_5:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR4_WRITE_POL_4:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR4_WRITE_POL_3:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR4_WRITE_POL_2:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR4_WRITE_POL_1:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR4_WRITE_POL_0:** B-Unit IMR4 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value. |

## 44.3.47 IMR5 Base (B_CR_BIMR5BASE_0_0_0_MCHBAR)—Offset 6910h

This register, along with IMR5MASK, IMR5RAC and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR5RAC and IMR5WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6910h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR5 Base (IMR5_BASE):** Specifies bits 38:10 of the start address of IMR5 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR5MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR5 defined region. |

## 44.3.48 IMR5 Mask (B_CR_BIMR5MASK_0_0_0_MCHBAR)—Offset 6914h

This register, along with IMR5BASE, IMR5RAC and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR5RAC and IMR5WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6914h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR5 Mask (IMR5_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR5BASE[28:0] value. A match indicates that the incoming address falls within the IMR5 region. |

## 44.3.49 IMR5 Control Policy (B_CR_BIMR5CP_0_0_0_MCHBAR)—Offset 6918h

This register controls the access policy to the Read Access Policy BIMR5RAC, the Write Access Policy BIMR5WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6918h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR5 Control Policy (IMR5_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.50 IMR5 Read Access Policy (B_CR_BIMR5RAC_0_0_0_MCHBAR)—Offset 6920h

This register, along with IMR5BASE, IMR5MASK and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR5. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6920h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR5 Read Access Policy 63 (IMR5_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR5 Read Access Policy 62 (IMR5_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR5 Read Access Policy 61 (IMR5_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR5 Read Access Policy 60 (IMR5_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR5 Read Access Policy 59 (IMR5_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR5 Read Access Policy 58 (IMR5_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR5 Read Access Policy 57 (IMR5_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR5 Read Access Policy 56 (IMR5_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR5 Read Access Policy 55 (IMR5_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR5 Read Access Policy 54 (IMR5_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR5 Read Access Policy 53 (IMR5_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR5 Read Access Policy 52 (IMR5_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR5 Read Access Policy 51 (IMR5_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR5 Read Access Policy 50 (IMR5_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR5 Read Access Policy 49 (IMR5_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR5 Read Access Policy 48 (IMR5_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR5 Read Access Policy 47 (IMR5_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR5 Read Access Policy 46 (IMR5_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR5 Read Access Policy 45 (IMR5_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR5 Read Access Policy 44 (IMR5_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR5 Read Access Policy 43 (IMR5_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR5 Read Access Policy 42 (IMR5_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR5 Read Access Policy 41 (IMR5_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR5 Read Access Policy 40 (IMR5_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR5 Read Access Policy 39 (IMR5_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR5 Read Access Policy 38 (IMR5_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR5 Read Access Policy 37 (IMR5_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR5 Read Access Policy 36 (IMR5_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR5 Read Access Policy 35 (IMR5_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR5 Read Access Policy 34 (IMR5_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR5 Read Access Policy 33 (IMR5_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR5 Read Access Policy 32 (IMR5_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR5 Read Access Policy 31 (IMR5_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR5 Read Access Policy 30 (IMR5_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR5 Read Access Policy 29 (IMR5_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR5 Read Access Policy 28 (IMR5_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR5 Read Access Policy 27 (IMR5_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR5 Read Access Policy 26 (IMR5_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR5 Read Access Policy 25 (IMR5_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR5 Read Access Policy 24 (IMR5_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR5 Read Access Policy 23 (IMR5_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR5 Read Access Policy 22 (IMR5_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR5 Read Access Policy 21 (IMR5_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR5 Read Access Policy 20 (IMR5_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR5 Read Access Policy 19 (IMR5_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR5 Read Access Policy 18 (IMR5_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR5 Read Access Policy 17 (IMR5_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR5 Read Access Policy 16 (IMR5_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR5 Read Access Policy 15 (IMR5_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR5 Read Access Policy 14 (IMR5_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR5 Read Access Policy 13 (IMR5_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR5 Read Access Policy 12 (IMR5_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR5 Read Access Policy 11 (IMR5_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR5 Read Access Policy 10 (IMR5_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR5 Read Access Policy 9 (IMR5_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR5 Read Access Policy 8 (IMR5_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR5 Read Access Policy 7 (IMR5_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR5 Read Access Policy 6 (IMR5_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR5 Read Access Policy 5 (IMR5_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR5 Read Access Policy 4 (IMR5_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR5 Read Access Policy 3 (IMR5_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR5 Read Access Policy 2 (IMR5_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR5 Read Access Policy 1 (IMR5_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR5 Read Access Policy 0 (IMR5_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

## 44.3.51 IMR5 Write Access Policy (B_CR_BIMR5WAC_0_0_0_MCHBAR)—Offset 6928h

This register, along with IMR5BASE, IMR5MASK and IMR5RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR5. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6928h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR5 Write Access Policy 63 (IMR5_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR5 Write Access Policy 62 (IMR5_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR5 Write Access Policy 61 (IMR5_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR5 Write Access Policy 60 (IMR5_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR5 Write Access Policy 59 (IMR5_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR5 Write Access Policy 58 (IMR5_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR5 Write Access Policy 57 (IMR5_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR5 Write Access Policy 56 (IMR5_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR5 Write Access Policy 55 (IMR5_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR5 Write Access Policy 54 (IMR5_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR5 Write Access Policy 53 (IMR5_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR5 Write Access Policy 52 (IMR5_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR5 Write Access Policy 51 (IMR5_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR5 Write Access Policy 50 (IMR5_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR5 Write Access Policy 49 (IMR5_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR5 Write Access Policy 48 (IMR5_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR5 Write Access Policy 47 (IMR5_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR5 Write Access Policy 46 (IMR5_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR5 Write Access Policy 45 (IMR5_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR5 Write Access Policy 44 (IMR5_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR5 Write Access Policy 43 (IMR5_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR5 Write Access Policy 42 (IMR5_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR5 Write Access Policy 41 (IMR5_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR5 Write Access Policy 40 (IMR5_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR5 Write Access Policy 39 (IMR5_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR5 Write Access Policy 38 (IMR5_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR5 Write Access Policy 37 (IMR5_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR5 Write Access Policy 36 (IMR5_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR5 Write Access Policy 35 (IMR5_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR5 Write Access Policy 34 (IMR5_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR5 Write Access Policy 33 (IMR5_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR5 Write Access Policy 32 (IMR5_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR5 Write Access Policy 31 (IMR5_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR5 Write Access Policy 30 (IMR5_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR5 Write Access Policy 29 (IMR5_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR5 Write Access Policy 28 (IMR5_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR5 Write Access Policy 27 (IMR5_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR5 Write Access Policy 26 (IMR5_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR5 Write Access Policy 25 (IMR5_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR5 Write Access Policy 24 (IMR5_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR5 Write Access Policy 23 (IMR5_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR5 Write Access Policy 22 (IMR5_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR5 Write Access Policy 21 (IMR5_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR5 Write Access Policy 20 (IMR5_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR5 Write Access Policy 19 (IMR5_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR5 Write Access Policy 18 (IMR5_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR5 Write Access Policy 17 (IMR5_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR5 Write Access Policy 16 (IMR5_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR5 Write Access Policy 15 (IMR5_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR5 Write Access Policy 14 (IMR5_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR5 Write Access Policy 13 (IMR5_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR5 Write Access Policy 12 (IMR5_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR5 Write Access Policy 11 (IMR5_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR5 Write Access Policy 10 (IMR5_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR5 Write Access Policy 9 (IMR5_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR5 Write Access Policy 8 (IMR5_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR5 Write Access Policy 7 (IMR5_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR5 Write Access Policy 6 (IMR5_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR5 Write Access Policy 5 (IMR5_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR5 Write Access Policy 4 (IMR5_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR5 Write Access Policy 3 (IMR5_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR5 Write Access Policy 2 (IMR5_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR5 Write Access Policy 1 (IMR5_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR5 Write Access Policy 0 (IMR5_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value. |

## 44.3.52 IMR6 Base (B_CR_BIMR6BASE_0_0_0_MCHBAR)—Offset 6930h

This register, along with IMR6MASK, IMR6RAC and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR6RAC and IMR6WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6930h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR6 Base (IMR6_BASE):** Specifies bits 38:10 of the start address of IMR6 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR6MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR6 defined region. |

## 44.3.53 IMR6 Mask (B_CR_BIMR6MASK_0_0_0_MCHBAR)—Offset 6934h

This register, along with IMR6BASE, IMR6RAC and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR6RAC and IMR6WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6934h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR6 Mask (IMR6_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR6BASE[28:0] value. A match indicates that the incoming address falls within the IMR6 region. |

## 44.3.54 IMR6 Control Policy (B_CR_BIMR6CP_0_0_0_MCHBAR)—Offset 6938h

This register controls the access policy to the Read Access Policy BIMR6RAC, the Write Access Policy BIMR6WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6938h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR6 Control Policy (IMR6_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC, BIMR6CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.55 IMR6 Read Access Policy (B_CR_BIMR6RAC_0_0_0_MCHBAR)—Offset 6940h

This register, along with IMR6BASE, IMR6MASK and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR6. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6940h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR6 Read Access Policy 63 (IMR6_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR6 Read Access Policy 62 (IMR6_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR6 Read Access Policy 61 (IMR6_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR6 Read Access Policy 60 (IMR6_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR6 Read Access Policy 59 (IMR6_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR6 Read Access Policy 58 (IMR6_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR6 Read Access Policy 57 (IMR6_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR6 Read Access Policy 56 (IMR6_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR6 Read Access Policy 55 (IMR6_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR6 Read Access Policy 54 (IMR6_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR6 Read Access Policy 53 (IMR6_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR6 Read Access Policy 52 (IMR6_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR6 Read Access Policy 51 (IMR6_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR6 Read Access Policy 50 (IMR6_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR6 Read Access Policy 49 (IMR6_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR6 Read Access Policy 48 (IMR6_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR6 Read Access Policy 47 (IMR6_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR6 Read Access Policy 46 (IMR6_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR6 Read Access Policy 45 (IMR6_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR6 Read Access Policy 44 (IMR6_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR6 Read Access Policy 43 (IMR6_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR6 Read Access Policy 42 (IMR6_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR6 Read Access Policy 41 (IMR6_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR6 Read Access Policy 40 (IMR6_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR6 Read Access Policy 39 (IMR6_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR6 Read Access Policy 38 (IMR6_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR6 Read Access Policy 37 (IMR6_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR6 Read Access Policy 36 (IMR6_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR6 Read Access Policy 35 (IMR6_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR6 Read Access Policy 34 (IMR6_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR6 Read Access Policy 33 (IMR6_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR6 Read Access Policy 32 (IMR6_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR6 Read Access Policy 31 (IMR6_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR6 Read Access Policy 30 (IMR6_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR6 Read Access Policy 29 (IMR6_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR6 Read Access Policy 28 (IMR6_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR6 Read Access Policy 27 (IMR6_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR6 Read Access Policy 26 (IMR6_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR6 Read Access Policy 25 (IMR6_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR6 Read Access Policy 24 (IMR6_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR6 Read Access Policy 23 (IMR6_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR6 Read Access Policy 22 (IMR6_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR6 Read Access Policy 21 (IMR6_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR6 Read Access Policy 20 (IMR6_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR6 Read Access Policy 19 (IMR6_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR6 Read Access Policy 18 (IMR6_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR6 Read Access Policy 17 (IMR6_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR6 Read Access Policy 16 (IMR6_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR6 Read Access Policy 15 (IMR6_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR6 Read Access Policy 14 (IMR6_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR6 Read Access Policy 13 (IMR6_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR6 Read Access Policy 12 (IMR6_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR6 Read Access Policy 11 (IMR6_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR6 Read Access Policy 10 (IMR6_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR6 Read Access Policy 9 (IMR6_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR6 Read Access Policy 8 (IMR6_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR6 Read Access Policy 7 (IMR6_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR6 Read Access Policy 6 (IMR6_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR6 Read Access Policy 5 (IMR6_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR6 Read Access Policy 4 (IMR6_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR6 Read Access Policy 3 (IMR6_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR6 Read Access Policy 2 (IMR6_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR6 Read Access Policy 1 (IMR6_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR6 Read Access Policy 0 (IMR6_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

## 44.3.56 IMR6 Write Access Policy (B_CR_BIMR6WAC_0_0_0_MCHBAR)—Offset 6948h

This register, along with IMR6BASE, IMR6MASK and IMR6RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR6. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6948h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR6 Write Access Policy 63 (IMR6_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR6 Write Access Policy 62 (IMR6_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR6 Write Access Policy 61 (IMR6_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR6 Write Access Policy 60 (IMR6_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR6 Write Access Policy 59 (IMR6_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR6 Write Access Policy 58 (IMR6_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR6 Write Access Policy 57 (IMR6_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR6 Write Access Policy 56 (IMR6_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR6 Write Access Policy 55 (IMR6_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR6 Write Access Policy 54 (IMR6_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR6 Write Access Policy 53 (IMR6_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR6 Write Access Policy 52 (IMR6_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR6 Write Access Policy 51 (IMR6_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR6 Write Access Policy 50 (IMR6_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR6 Write Access Policy 49 (IMR6_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR6 Write Access Policy 48 (IMR6_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR6 Write Access Policy 47 (IMR6_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR6 Write Access Policy 46 (IMR6_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR6 Write Access Policy 45 (IMR6_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR6 Write Access Policy 44 (IMR6_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR6 Write Access Policy 43 (IMR6_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR6 Write Access Policy 42 (IMR6_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR6 Write Access Policy 41 (IMR6_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR6 Write Access Policy 40 (IMR6_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR6 Write Access Policy 39 (IMR6_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR6 Write Access Policy 38 (IMR6_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR6 Write Access Policy 37 (IMR6_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR6 Write Access Policy 36 (IMR6_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR6 Write Access Policy 35 (IMR6_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR6 Write Access Policy 34 (IMR6_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR6 Write Access Policy 33 (IMR6_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR6 Write Access Policy 32 (IMR6_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR6 Write Access Policy 31 (IMR6_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR6 Write Access Policy 30 (IMR6_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR6 Write Access Policy 29 (IMR6_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR6 Write Access Policy 28 (IMR6_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR6 Write Access Policy 27 (IMR6_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR6 Write Access Policy 26 (IMR6_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR6 Write Access Policy 25 (IMR6_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR6 Write Access Policy 24 (IMR6_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR6 Write Access Policy 23 (IMR6_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR6 Write Access Policy 22 (IMR6_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR6 Write Access Policy 21 (IMR6_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR6 Write Access Policy 20 (IMR6_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR6 Write Access Policy 19 (IMR6_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR6 Write Access Policy 18 (IMR6_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR6 Write Access Policy 17 (IMR6_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR6 Write Access Policy 16 (IMR6_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR6 Write Access Policy 15 (IMR6_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR6 Write Access Policy 14 (IMR6_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR6 Write Access Policy 13 (IMR6_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR6 Write Access Policy 12 (IMR6_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR6 Write Access Policy 11 (IMR6_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR6 Write Access Policy 10 (IMR6_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR6 Write Access Policy 9 (IMR6_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR6 Write Access Policy 8 (IMR6_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR6 Write Access Policy 7 (IMR6_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR6 Write Access Policy 6 (IMR6_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR6 Write Access Policy 5 (IMR6_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR6 Write Access Policy 4 (IMR6_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR6 Write Access Policy 3 (IMR6_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR6 Write Access Policy 2 (IMR6_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR6 Write Access Policy 1 (IMR6_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR6 Write Access Policy 0 (IMR6_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value. |

## 44.3.57 IMR7 Base (B_CR_BIMR7BASE_0_0_0_MCHBAR)—Offset 6950h

This register, along with IMR7MASK, IMR7RAC and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR7RAC and IMR7WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6950h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR7 Base (IMR7_BASE):** Specifies bits 38:10 of the start address of IMR7 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR7MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR7 defined region. |

## 44.3.58 IMR7 Mask (B_CR_BIMR7MASK_0_0_0_MCHBAR)—Offset 6954h

This register, along with IMR7BASE, IMR7RAC and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR7RAC and IMR7WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6954h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR7 Mask (IMR7_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR7BASE[28:0] value. A match indicates that the incoming address falls within the IMR7 region. |

## 44.3.59 IMR7 Control Policy (B_CR_BIMR7CP_0_0_0_MCHBAR)—Offset 6958h

This register controls the access policy to the Read Access Policy BIMR7RAC, the Write Access Policy BIMR7WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6958h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR7 Control Policy (IMR7_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.60 IMR7 Read Access Policy (B_CR_BIMR7RAC_0_0_0_MCHBAR)—Offset 6960h

This register, along with IMR7BASE, IMR7MASK and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR7. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6960h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR7 Read Access Policy 63 (IMR7_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR7 Read Access Policy 62 (IMR7_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR7 Read Access Policy 61 (IMR7_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR7 Read Access Policy 60 (IMR7_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR7 Read Access Policy 59 (IMR7_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR7 Read Access Policy 58 (IMR7_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR7 Read Access Policy 57 (IMR7_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR7 Read Access Policy 56 (IMR7_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR7 Read Access Policy 55 (IMR7_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR7 Read Access Policy 54 (IMR7_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR7 Read Access Policy 53 (IMR7_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR7 Read Access Policy 52 (IMR7_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR7 Read Access Policy 51 (IMR7_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR7 Read Access Policy 50 (IMR7_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR7 Read Access Policy 49 (IMR7_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR7 Read Access Policy 48 (IMR7_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR7 Read Access Policy 47 (IMR7_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR7 Read Access Policy 46 (IMR7_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR7 Read Access Policy 45 (IMR7_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR7 Read Access Policy 44 (IMR7_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR7 Read Access Policy 43 (IMR7_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR7 Read Access Policy 42 (IMR7_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR7 Read Access Policy 41 (IMR7_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR7 Read Access Policy 40 (IMR7_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR7 Read Access Policy 39 (IMR7_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR7 Read Access Policy 38 (IMR7_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR7 Read Access Policy 37 (IMR7_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR7 Read Access Policy 36 (IMR7_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR7 Read Access Policy 35 (IMR7_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR7 Read Access Policy 34 (IMR7_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR7 Read Access Policy 33 (IMR7_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR7 Read Access Policy 32 (IMR7_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR7 Read Access Policy 31 (IMR7_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR7 Read Access Policy 30 (IMR7_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR7 Read Access Policy 29 (IMR7_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR7 Read Access Policy 28 (IMR7_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR7 Read Access Policy 27 (IMR7_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR7 Read Access Policy 26 (IMR7_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR7 Read Access Policy 25 (IMR7_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR7 Read Access Policy 24 (IMR7_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR7 Read Access Policy 23 (IMR7_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR7 Read Access Policy 22 (IMR7_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR7 Read Access Policy 21 (IMR7_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR7 Read Access Policy 20 (IMR7_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR7 Read Access Policy 19 (IMR7_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR7 Read Access Policy 18 (IMR7_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR7 Read Access Policy 17 (IMR7_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR7 Read Access Policy 16 (IMR7_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR7 Read Access Policy 15 (IMR7_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR7 Read Access Policy 14 (IMR7_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR7 Read Access Policy 13 (IMR7_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR7 Read Access Policy 12 (IMR7_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR7 Read Access Policy 11 (IMR7_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR7 Read Access Policy 10 (IMR7_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR7 Read Access Policy 9 (IMR7_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR7 Read Access Policy 8 (IMR7_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR7 Read Access Policy 7 (IMR7_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR7 Read Access Policy 6 (IMR7_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR7 Read Access Policy 5 (IMR7_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR7 Read Access Policy 4 (IMR7_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR7 Read Access Policy 3 (IMR7_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR7 Read Access Policy 2 (IMR7_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR7 Read Access Policy 1 (IMR7_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR7 Read Access Policy 0 (IMR7_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

## 44.3.61 IMR7 Write Access Policy (B_CR_BIMR7WAC_0_0_0_MCHBAR)—Offset 6968h

This register, along with IMR7BASE, IMR7MASK and IMR7RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR7. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6968h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR7 Write Access Policy 63 (IMR7_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR7 Write Access Policy 62 (IMR7_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR7 Write Access Policy 61 (IMR7_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR7 Write Access Policy 60 (IMR7_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR7 Write Access Policy 59 (IMR7_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR7 Write Access Policy 58 (IMR7_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR7 Write Access Policy 57 (IMR7_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR7 Write Access Policy 56 (IMR7_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR7 Write Access Policy 55 (IMR7_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR7 Write Access Policy 54 (IMR7_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR7 Write Access Policy 53 (IMR7_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR7 Write Access Policy 52 (IMR7_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR7 Write Access Policy 51 (IMR7_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR7 Write Access Policy 50 (IMR7_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR7 Write Access Policy 49 (IMR7_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR7 Write Access Policy 48 (IMR7_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR7 Write Access Policy 47 (IMR7_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR7 Write Access Policy 46 (IMR7_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR7 Write Access Policy 45 (IMR7_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR7 Write Access Policy 44 (IMR7_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR7 Write Access Policy 43 (IMR7_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR7 Write Access Policy 42 (IMR7_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR7 Write Access Policy 41 (IMR7_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR7 Write Access Policy 40 (IMR7_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR7 Write Access Policy 39 (IMR7_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR7 Write Access Policy 38 (IMR7_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR7 Write Access Policy 37 (IMR7_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR7 Write Access Policy 36 (IMR7_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR7 Write Access Policy 35 (IMR7_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR7 Write Access Policy 34 (IMR7_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR7 Write Access Policy 33 (IMR7_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR7 Write Access Policy 32 (IMR7_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR7 Write Access Policy 31 (IMR7_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR7 Write Access Policy 30 (IMR7_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR7 Write Access Policy 29 (IMR7_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR7 Write Access Policy 28 (IMR7_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR7 Write Access Policy 27 (IMR7_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR7 Write Access Policy 26 (IMR7_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR7 Write Access Policy 25 (IMR7_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR7 Write Access Policy 24 (IMR7_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR7 Write Access Policy 23 (IMR7_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR7 Write Access Policy 22 (IMR7_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR7 Write Access Policy 21 (IMR7_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR7 Write Access Policy 20 (IMR7_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR7 Write Access Policy 19 (IMR7_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR7 Write Access Policy 18 (IMR7_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR7 Write Access Policy 17 (IMR7_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR7 Write Access Policy 16 (IMR7_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR7 Write Access Policy 15 (IMR7_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR7 Write Access Policy 14 (IMR7_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR7 Write Access Policy 13 (IMR7_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR7 Write Access Policy 12 (IMR7_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR7 Write Access Policy 11 (IMR7_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR7 Write Access Policy 10 (IMR7_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR7 Write Access Policy 9 (IMR7_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR7 Write Access Policy 8 (IMR7_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR7 Write Access Policy 7 (IMR7_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR7 Write Access Policy 6 (IMR7_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR7 Write Access Policy 5 (IMR7_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR7 Write Access Policy 4 (IMR7_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR7 Write Access Policy 3 (IMR7_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR7 Write Access Policy 2 (IMR7_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR7 Write Access Policy 1 (IMR7_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR7 Write Access Policy 0 (IMR7_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value. |

## 44.3.62 IMR8 Base (B_CR_BIMR8BASE_0_0_0_MCHBAR)—Offset 6970h

This register, along with IMR8MASK, IMR8RAC and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR8RAC and IMR8WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6970h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR8 Base (IMR8_BASE):** Specifies bits 38:10 of the start address of IMR8 region. IMR region size must be a strict poweroftwo at least 1KB and naturally aligned to the size. These bits are compared with the result of the IMR8MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR8 defined region. |

## 44.3.63 IMR8 Mask (B_CR_BIMR8MASK_0_0_0_MCHBAR)—Offset 6974h

This register, along with IMR8BASE, IMR8RAC and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR8RAC and IMR8WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6974h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR8 Mask (IMR8_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR8BASE[28:0] value. A match indicates that the incoming address falls within the IMR8 region. |

## 44.3.64 IMR8 Control Policy (B_CR_BIMR8CP_0_0_0_MCHBAR)—Offset 6978h

This register controls the access policy to the Read Access Policy BIMR8RAC, the Write Access Policy BIMR8WAC, and, self-referentially to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6978h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR8 Control Policy (IMR8_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.65 IMR8 Read Access Policy (B_CR_BIMR8RAC_0_0_0_MCHBAR)—Offset 6980h

This register, along with IMR8BASE, IMR8MASK and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR8. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6980h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR8 Read Access Policy 63 (IMR8_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR8 Read Access Policy 62 (IMR8_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR8 Read Access Policy 61 (IMR8_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR8 Read Access Policy 60 (IMR8_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR8 Read Access Policy 59 (IMR8_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR8 Read Access Policy 58 (IMR8_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR8 Read Access Policy 57 (IMR8_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR8 Read Access Policy 56 (IMR8_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR8 Read Access Policy 55 (IMR8_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR8 Read Access Policy 54 (IMR8_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR8 Read Access Policy 53 (IMR8_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR8 Read Access Policy 52 (IMR8_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR8 Read Access Policy 51 (IMR8_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR8 Read Access Policy 50 (IMR8_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR8 Read Access Policy 49 (IMR8_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR8 Read Access Policy 48 (IMR8_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR8 Read Access Policy 47 (IMR8_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR8 Read Access Policy 46 (IMR8_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR8 Read Access Policy 45 (IMR8_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR8 Read Access Policy 44 (IMR8_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR8 Read Access Policy 43 (IMR8_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR8 Read Access Policy 42 (IMR8_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR8 Read Access Policy 41 (IMR8_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR8 Read Access Policy 40 (IMR8_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR8 Read Access Policy 39 (IMR8_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR8 Read Access Policy 38 (IMR8_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR8 Read Access Policy 37 (IMR8_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR8 Read Access Policy 36 (IMR8_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR8 Read Access Policy 35 (IMR8_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR8 Read Access Policy 34 (IMR8_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR8 Read Access Policy 33 (IMR8_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR8 Read Access Policy 32 (IMR8_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR8 Read Access Policy 31 (IMR8_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR8 Read Access Policy 30 (IMR8_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR8 Read Access Policy 29 (IMR8_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR8 Read Access Policy 28 (IMR8_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR8 Read Access Policy 27 (IMR8_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR8 Read Access Policy 26 (IMR8_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR8 Read Access Policy 25 (IMR8_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR8 Read Access Policy 24 (IMR8_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR8 Read Access Policy 23 (IMR8_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR8 Read Access Policy 22 (IMR8_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR8 Read Access Policy 21 (IMR8_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR8 Read Access Policy 20 (IMR8_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR8 Read Access Policy 19 (IMR8_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR8 Read Access Policy 18 (IMR8_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR8 Read Access Policy 17 (IMR8_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR8 Read Access Policy 16 (IMR8_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR8 Read Access Policy 15 (IMR8_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR8 Read Access Policy 14 (IMR8_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR8 Read Access Policy 13 (IMR8_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR8 Read Access Policy 12 (IMR8_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR8 Read Access Policy 11 (IMR8_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR8 Read Access Policy 10 (IMR8_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR8 Read Access Policy 9 (IMR8_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR8 Read Access Policy 8 (IMR8_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR8 Read Access Policy 7 (IMR8_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR8 Read Access Policy 6 (IMR8_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR8 Read Access Policy 5 (IMR8_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR8 Read Access Policy 4 (IMR8_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR8 Read Access Policy 3 (IMR8_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR8 Read Access Policy 2 (IMR8_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR8 Read Access Policy 1 (IMR8_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR8 Read Access Policy 0 (IMR8_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

## 44.3.66 IMR8 Write Access Policy (B_CR_BIMR8WAC_0_0_0_MCHBAR)—Offset 6988h

This register, along with IMR8BASE, IMR8MASK and IMR8RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR8. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6988h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR8 Write Access Policy 63 (IMR8_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR8 Write Access Policy 62 (IMR8_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR8 Write Access Policy 61 (IMR8_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR8 Write Access Policy 60 (IMR8_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR8 Write Access Policy 59 (IMR8_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR8 Write Access Policy 58 (IMR8_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR8 Write Access Policy 57 (IMR8_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR8 Write Access Policy 56 (IMR8_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR8 Write Access Policy 55 (IMR8_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR8 Write Access Policy 54 (IMR8_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR8 Write Access Policy 53 (IMR8_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR8 Write Access Policy 52 (IMR8_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR8 Write Access Policy 51 (IMR8_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR8 Write Access Policy 50 (IMR8_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR8 Write Access Policy 49 (IMR8_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR8 Write Access Policy 48 (IMR8_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR8 Write Access Policy 47 (IMR8_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR8 Write Access Policy 46 (IMR8_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR8 Write Access Policy 45 (IMR8_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR8 Write Access Policy 44 (IMR8_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR8 Write Access Policy 43 (IMR8_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR8 Write Access Policy 42 (IMR8_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR8 Write Access Policy 41 (IMR8_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR8 Write Access Policy 40 (IMR8_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR8 Write Access Policy 39 (IMR8_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR8 Write Access Policy 38 (IMR8_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR8 Write Access Policy 37 (IMR8_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR8 Write Access Policy 36 (IMR8_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR8 Write Access Policy 35 (IMR8_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR8 Write Access Policy 34 (IMR8_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR8 Write Access Policy 33 (IMR8_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR8 Write Access Policy 32 (IMR8_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR8 Write Access Policy 31 (IMR8_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR8 Write Access Policy 30 (IMR8_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR8 Write Access Policy 29 (IMR8_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR8 Write Access Policy 28 (IMR8_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR8 Write Access Policy 27 (IMR8_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR8 Write Access Policy 26 (IMR8_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR8 Write Access Policy 25 (IMR8_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR8 Write Access Policy 24 (IMR8_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR8 Write Access Policy 23 (IMR8_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR8 Write Access Policy 22 (IMR8_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR8 Write Access Policy 21 (IMR8_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR8 Write Access Policy 20 (IMR8_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR8 Write Access Policy 19 (IMR8_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR8 Write Access Policy 18 (IMR8_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR8 Write Access Policy 17 (IMR8_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR8 Write Access Policy 16 (IMR8_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR8 Write Access Policy 15 (IMR8_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR8 Write Access Policy 14 (IMR8_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR8 Write Access Policy 13 (IMR8_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR8 Write Access Policy 12 (IMR8_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR8 Write Access Policy 11 (IMR8_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR8 Write Access Policy 10 (IMR8_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR8 Write Access Policy 9 (IMR8_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR8 Write Access Policy 8 (IMR8_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR8 Write Access Policy 7 (IMR8_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR8 Write Access Policy 6 (IMR8_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR8 Write Access Policy 5 (IMR8_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR8 Write Access Policy 4 (IMR8_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR8 Write Access Policy 3 (IMR8_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR8 Write Access Policy 2 (IMR8_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR8 Write Access Policy 1 (IMR8_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR8 Write Access Policy 0 (IMR8_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value. |

## 44.3.67 IMR9 Base (B_CR_BIMR9BASE_0_0_0_MCHBAR)—Offset 6990h

This register, along with IMR9MASK, IMR9RAC and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR9RAC and IMR9WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6990h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 1 IMR9 Base (IMR9_BASE):** Specifies bits 38:10 of the start address of IMR9 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR9MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR9 defined region. |

## 44.3.68 IMR9 Mask (B_CR_BIMR9MASK_0_0_0_MCHBAR)—Offset 6994h

This register, along with IMR9BASE, IMR9RAC and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR9RAC and IMR9WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6994h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 1 IMR9 Mask (IMR9_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR9BASE[28:0] value. A match indicates that the incoming address falls within the IMR9 region. |

## 44.3.69 IMR9 Control Policy (B_CR_BIMR9CP_0_0_0_MCHBAR)—Offset 6998h

This register controls the access policy to the Read Access Policy BIMR9RAC, the Write Access Policy BIMR9WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6998h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR9 Control Policy (IMR9_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC, BIMR9CP registers based on the value from each agent's 6bit SAI field. |

## 44.3.70 IMR9 Read Access Policy (B_CR_BIMR9RAC_0_0_0_MCHBAR)—Offset 69A0h

This register, along with IMR9BASE, IMR9MASK and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR9. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69A0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR9 Read Access Policy 63 (IMR9_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR9 Read Access Policy 62 (IMR9_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR9 Read Access Policy 61 (IMR9_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR9 Read Access Policy 60 (IMR9_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR9 Read Access Policy 59 (IMR9_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR9 Read Access Policy 58 (IMR9_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR9 Read Access Policy 57 (IMR9_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR9 Read Access Policy 56 (IMR9_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR9 Read Access Policy 55 (IMR9_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR9 Read Access Policy 54 (IMR9_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR9 Read Access Policy 53 (IMR9_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR9 Read Access Policy 52 (IMR9_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR9 Read Access Policy 51 (IMR9_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR9 Read Access Policy 50 (IMR9_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR9 Read Access Policy 49 (IMR9_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR9 Read Access Policy 48 (IMR9_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR9 Read Access Policy 47 (IMR9_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR9 Read Access Policy 46 (IMR9_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR9 Read Access Policy 45 (IMR9_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR9 Read Access Policy 44 (IMR9_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR9 Read Access Policy 43 (IMR9_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR9 Read Access Policy 42 (IMR9_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR9 Read Access Policy 41 (IMR9_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR9 Read Access Policy 40 (IMR9_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR9 Read Access Policy 39 (IMR9_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR9 Read Access Policy 38 (IMR9_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR9 Read Access Policy 37 (IMR9_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR9 Read Access Policy 36 (IMR9_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR9 Read Access Policy 35 (IMR9_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR9 Read Access Policy 34 (IMR9_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR9 Read Access Policy 33 (IMR9_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR9 Read Access Policy 32 (IMR9_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR9 Read Access Policy 31 (IMR9_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR9 Read Access Policy 30 (IMR9_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR9 Read Access Policy 29 (IMR9_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR9 Read Access Policy 28 (IMR9_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR9 Read Access Policy 27 (IMR9_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR9 Read Access Policy 26 (IMR9_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR9 Read Access Policy 25 (IMR9_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR9 Read Access Policy 24 (IMR9_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR9 Read Access Policy 23 (IMR9_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR9 Read Access Policy 22 (IMR9_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR9 Read Access Policy 21 (IMR9_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR9 Read Access Policy 20 (IMR9_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR9 Read Access Policy 19 (IMR9_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR9 Read Access Policy 18 (IMR9_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR9 Read Access Policy 17 (IMR9_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR9 Read Access Policy 16 (IMR9_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR9 Read Access Policy 15 (IMR9_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR9 Read Access Policy 14 (IMR9_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR9 Read Access Policy 13 (IMR9_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR9 Read Access Policy 12 (IMR9_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR9 Read Access Policy 11 (IMR9_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR9 Read Access Policy 10 (IMR9_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR9 Read Access Policy 9 (IMR9_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR9 Read Access Policy 8 (IMR9_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR9 Read Access Policy 7 (IMR9_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR9 Read Access Policy 6 (IMR9_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR9 Read Access Policy 5 (IMR9_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR9 Read Access Policy 4 (IMR9_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR9 Read Access Policy 3 (IMR9_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR9 Read Access Policy 2 (IMR9_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR9 Read Access Policy 1 (IMR9_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR9 Read Access Policy 0 (IMR9_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

## 44.3.71 IMR9 Write Access Policy (B_CR_BIMR9WAC_0_0_0_MCHBAR)—Offset 69A8h

This register, along with IMR9BASE, IMR9MASK and IMR9RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR9. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69A8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR9 Write Access Policy 63 (IMR9_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR9 Write Access Policy 62 (IMR9_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR9 Write Access Policy 61 (IMR9_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR9 Write Access Policy 60 (IMR9_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR9 Write Access Policy 59 (IMR9_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR9 Write Access Policy 58 (IMR9_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR9 Write Access Policy 57 (IMR9_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR9 Write Access Policy 56 (IMR9_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR9 Write Access Policy 55 (IMR9_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR9 Write Access Policy 54 (IMR9_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR9 Write Access Policy 53 (IMR9_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR9 Write Access Policy 52 (IMR9_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR9 Write Access Policy 51 (IMR9_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR9 Write Access Policy 50 (IMR9_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR9 Write Access Policy 49 (IMR9_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR9 Write Access Policy 48 (IMR9_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR9 Write Access Policy 47 (IMR9_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR9 Write Access Policy 46 (IMR9_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR9 Write Access Policy 45 (IMR9_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR9 Write Access Policy 44 (IMR9_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR9 Write Access Policy 43 (IMR9_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR9 Write Access Policy 42 (IMR9_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR9 Write Access Policy 41 (IMR9_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR9 Write Access Policy 40 (IMR9_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR9 Write Access Policy 39 (IMR9_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR9 Write Access Policy 38 (IMR9_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR9 Write Access Policy 37 (IMR9_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR9 Write Access Policy 36 (IMR9_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR9 Write Access Policy 35 (IMR9_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR9 Write Access Policy 34 (IMR9_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR9 Write Access Policy 33 (IMR9_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR9 Write Access Policy 32 (IMR9_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR9 Write Access Policy 31 (IMR9_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR9 Write Access Policy 30 (IMR9_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR9 Write Access Policy 29 (IMR9_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR9 Write Access Policy 28 (IMR9_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR9 Write Access Policy 27 (IMR9_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR9 Write Access Policy 26 (IMR9_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR9 Write Access Policy 25 (IMR9_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR9 Write Access Policy 24 (IMR9_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR9 Write Access Policy 23 (IMR9_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR9 Write Access Policy 22 (IMR9_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR9 Write Access Policy 21 (IMR9_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR9 Write Access Policy 20 (IMR9_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR9 Write Access Policy 19 (IMR9_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR9 Write Access Policy 18 (IMR9_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR9 Write Access Policy 17 (IMR9_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR9 Write Access Policy 16 (IMR9_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR9 Write Access Policy 15 (IMR9_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR9 Write Access Policy 14 (IMR9_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR9 Write Access Policy 13 (IMR9_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR9 Write Access Policy 12 (IMR9_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR9 Write Access Policy 11 (IMR9_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR9 Write Access Policy 10 (IMR9_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR9 Write Access Policy 9 (IMR9_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR9 Write Access Policy 8 (IMR9_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR9 Write Access Policy 7 (IMR9_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR9 Write Access Policy 6 (IMR9_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR9 Write Access Policy 5 (IMR9_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR9 Write Access Policy 4 (IMR9_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR9 Write Access Policy 3 (IMR9_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR9 Write Access Policy 2 (IMR9_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR9 Write Access Policy 1 (IMR9_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR9 Write Access Policy 0 (IMR9_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value. |

## 44.3.72 IMR10 Base (B_CR_BIMR10BASE_0_0_0_MCHBAR)—Offset 69B0h

This register, along with IMR10MASK, IMR10RAC and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR10RAC and IMR10WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69B0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR10 Base (IMR10_BASE):** Specifies bits 38:10 of the start address of IMR10 region. IMR region size must be a strict poweroftwo at least 1KB and naturally aligned to the size. These bits are compared with the result of the IMR10MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR10 defined region. |

## 44.3.73 B_CR_BIMR10MASK_0_0_0_MCHBAR—Offset 69B4h

This register, along with IMR10BASE, IMR10RAC and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR10RAC and IMR10WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69B4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR10 Mask (IMR10_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR10BASE[28:0] value. A match indicates that the incoming address falls within the IMR10 region. |

## 44.3.74 IMR10 Control Policy (B_CR_BIMR10CP_0_0_0_MCHBAR)—Offset 69B8h

This register controls the access policy to the Read Access Policy BIMR10RAC, the Write Access Policy BIMR10WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69B8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR10 Control Policy (IMR10_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP, registers based on the value from each agent's 6bit SAI field. |

## 44.3.75 IMR10 Read Access Policy (B_CR_BIMR10RAC_0_0_0_MCHBAR)—Offset 69C0h

This register, along with IMR10BASE, IMR10MASK and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR10. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69C0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR10 Read Access Policy 63 (IMR10_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR10 Read Access Policy 62 (IMR10_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR10 Read Access Policy 61 (IMR10_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR10 Read Access Policy 60 (IMR10_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR10 Read Access Policy 59 (IMR10_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR10 Read Access Policy 58 (IMR10_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR10 Read Access Policy 57 (IMR10_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR10 Read Access Policy 56 (IMR10_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR10 Read Access Policy 55 (IMR10_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR10 Read Access Policy 54 (IMR10_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR10 Read Access Policy 53 (IMR10_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR10 Read Access Policy 52 (IMR10_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR10 Read Access Policy 51 (IMR10_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR10 Read Access Policy 50 (IMR10_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR10 Read Access Policy 49 (IMR10_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR10 Read Access Policy 48 (IMR10_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR10 Read Access Policy 47 (IMR10_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR10 Read Access Policy 46 (IMR10_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR10 Read Access Policy 45 (IMR10_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR10 Read Access Policy 44 (IMR10_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR10 Read Access Policy 43 (IMR10_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR10 Read Access Policy 42 (IMR10_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR10 Read Access Policy 41 (IMR10_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR10 Read Access Policy 40 (IMR10_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR10 Read Access Policy 39 (IMR10_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR10 Read Access Policy 38 (IMR10_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR10 Read Access Policy 37 (IMR10_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR10 Read Access Policy 36 (IMR10_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR10 Read Access Policy 35 (IMR10_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR10 Read Access Policy 34 (IMR10_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR10 Read Access Policy 33 (IMR10_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR10 Read Access Policy 32 (IMR10_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR10 Read Access Policy 31 (IMR10_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR10 Read Access Policy 30 (IMR10_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR10 Read Access Policy 29 (IMR10_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR10 Read Access Policy 28 (IMR10_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR10 Read Access Policy 27 (IMR10_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR10 Read Access Policy 26 (IMR10_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR10 Read Access Policy 25 (IMR10_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR10 Read Access Policy 24 (IMR10_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR10 Read Access Policy 23 (IMR10_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR10 Read Access Policy 22 (IMR10_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR10 Read Access Policy 21 (IMR10_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR10 Read Access Policy 20 (IMR10_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR10 Read Access Policy 19 (IMR10_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR10 Read Access Policy 18 (IMR10_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR10 Read Access Policy 17 (IMR10_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR10 Read Access Policy 16 (IMR10_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR10 Read Access Policy 15 (IMR10_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR10 Read Access Policy 14 (IMR10_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR10 Read Access Policy 13 (IMR10_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR10 Read Access Policy 12 (IMR10_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR10 Read Access Policy 11 (IMR10_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR10 Read Access Policy 10 (IMR10_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR10 Read Access Policy 9 (IMR10_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR10 Read Access Policy 8 (IMR10_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR10 Read Access Policy 7 (IMR10_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR10 Read Access Policy 6 (IMR10_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR10 Read Access Policy 5 (IMR10_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR10 Read Access Policy 4 (IMR10_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR10 Read Access Policy 3 (IMR10_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR10 Read Access Policy 2 (IMR10_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR10 Read Access Policy 1 (IMR10_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR10 Read Access Policy 0 (IMR10_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

## 44.3.76 IMR10 Write Access Policy (B_CR_BIMR10WAC_0_0_0_MCHBAR)—Offset 69C8h

This register along with IMR10BASE, IMR10MASK and IMR10RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR10. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69C8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0 RW | **IMR10 Write Access Policy 63 (IMR10_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0 RO | **IMR10 Write Access Policy 62 (IMR10_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0 RO | **IMR10 Write Access Policy 61 (IMR10_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0 RO | **IMR10 Write Access Policy 60 (IMR10_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0 RO | **IMR10 Write Access Policy 59 (IMR10_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0 RO | **IMR10 Write Access Policy 58 (IMR10_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0 RO | **IMR10 Write Access Policy 57 (IMR10_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0 RW | **IMR10 Write Access Policy 56 (IMR10_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0 RW | **IMR10 Write Access Policy 55 (IMR10_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0 RW | **IMR10 Write Access Policy 54 (IMR10_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0 RO | **IMR10 Write Access Policy 53 (IMR10_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0 RO | **IMR10 Write Access Policy 52 (IMR10_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0 RO | **IMR10 Write Access Policy 51 (IMR10_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0 RW | **IMR10 Write Access Policy 50 (IMR10_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0 RW | **IMR10 Write Access Policy 49 (IMR10_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0 RW | **IMR10 Write Access Policy 48 (IMR10_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0 RO | **IMR10 Write Access Policy 47 (IMR10_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0 RO | **IMR10 Write Access Policy 46 (IMR10_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR10 Write Access Policy 45 (IMR10_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR10 Write Access Policy 44 (IMR10_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR10 Write Access Policy 43 (IMR10_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR10 Write Access Policy 42 (IMR10_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR10 Write Access Policy 41 (IMR10_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR10 Write Access Policy 40 (IMR10_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR10 Write Access Policy 39 (IMR10_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR10 Write Access Policy 38 (IMR10_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR10 Write Access Policy 37 (IMR10_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR10 Write Access Policy 36 (IMR10_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR10 Write Access Policy 35 (IMR10_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR10 Write Access Policy 34 (IMR10_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR10 Write Access Policy 33 (IMR10_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR10 Write Access Policy 32 (IMR10_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR10 Write Access Policy 31 (IMR10_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR10 Write Access Policy 30 (IMR10_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR10 Write Access Policy 29 (IMR10_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR10 Write Access Policy 28 (IMR10_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR10 Write Access Policy 27 (IMR10_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR10 Write Access Policy 26 (IMR10_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR10 Write Access Policy 25 (IMR10_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR10 Write Access Policy 24 (IMR10_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR10 Write Access Policy 23 (IMR10_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR10 Write Access Policy 22 (IMR10_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR10 Write Access Policy 21 (IMR10_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR10 Write Access Policy 20 (IMR10_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR10 Write Access Policy 19 (IMR10_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR10 Write Access Policy 18 (IMR10_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR10 Write Access Policy 17 (IMR10_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR10 Write Access Policy 16 (IMR10_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR10 Write Access Policy 15 (IMR10_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR10 Write Access Policy 14 (IMR10_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR10 Write Access Policy 13 (IMR10_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR10 Write Access Policy 12 (IMR10_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR10 Write Access Policy 11 (IMR10_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR10 Write Access Policy 10 (IMR10_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR10 Write Access Policy 9 (IMR10_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR10 Write Access Policy 8 (IMR10_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR10 Write Access Policy 7 (IMR10_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR10 Write Access Policy 6 (IMR10_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR10 Write Access Policy 5 (IMR10_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR10 Write Access Policy 4 (IMR10_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR10 Write Access Policy 3 (IMR10_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR10 Write Access Policy 2 (IMR10_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR10 Write Access Policy 1 (IMR10_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR10 Write Access Policy 0 (IMR10_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value. |

## 44.3.77 IMR11 Base (B_CR_BIMR11BASE_0_0_0_MCHBAR)—Offset 69D0h

This register, along with IMR11MASK, IMR11RAC and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR11RAC and IMR11WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69D0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR11 Base (IMR11_BASE):** Specifies bits 38:10 of the start address of IMR11 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR11MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR11 defined region. |

## 44.3.78 IMR11 Mask (B_CR_BIMR11MASK_0_0_0_MCHBAR)—Offset 69D4h

This register, along with IMR11BASE, IMR11RAC and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR11RAC and IMR11WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69D4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR11 Mask (IMR11_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR11BASE[28:0] value. A match indicates that the incoming address falls within the IMR11 region. |

## 44.3.79 IMR11 Control Policy (B_CR_BIMR11CP_0_0_0_MCHBAR)—Offset 69D8h

This register controls the access policy to the Read Access Policy BIMR11RAC, the Write Access Policy BIMR11WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69D8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR11 Control Policy (IMR11_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.80 IMR11 Read Access Policy (B_CR_BIMR11RAC_0_0_0_MCHBAR)—Offset 69E0h

This register, along with IMR11BASE, IMR11MASK and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR11. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69E0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR11 Read Access Policy 63 (IMR11_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR11 Read Access Policy 62 (IMR11_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR11 Read Access Policy 61 (IMR11_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR11 Read Access Policy 60 (IMR11_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR11 Read Access Policy 59 (IMR11_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR11 Read Access Policy 58 (IMR11_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR11 Read Access Policy 57 (IMR11_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR11 Read Access Policy 56 (IMR11_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR11 Read Access Policy 55 (IMR11_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR11 Read Access Policy 54 (IMR11_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR11 Read Access Policy 53 (IMR11_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR11 Read Access Policy 52 (IMR11_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR11 Read Access Policy 51 (IMR11_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR11 Read Access Policy 50 (IMR11_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR11 Read Access Policy 49 (IMR11_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR11 Read Access Policy 48 (IMR11_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR11 Read Access Policy 47 (IMR11_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR11 Read Access Policy 46 (IMR11_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR11 Read Access Policy 45 (IMR11_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR11 Read Access Policy 44 (IMR11_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR11 Read Access Policy 43 (IMR11_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR11 Read Access Policy 42 (IMR11_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR11 Read Access Policy 41 (IMR11_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR11 Read Access Policy 40 (IMR11_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR11 Read Access Policy 39 (IMR11_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR11 Read Access Policy 38 (IMR11_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR11 Read Access Policy 37 (IMR11_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR11 Read Access Policy 36 (IMR11_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR11 Read Access Policy 35 (IMR11_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR11 Read Access Policy 34 (IMR11_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR11 Read Access Policy 33 (IMR11_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR11 Read Access Policy 32 (IMR11_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR11 Read Access Policy 31 (IMR11_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR11 Read Access Policy 30 (IMR11_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR11 Read Access Policy 29 (IMR11_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR11 Read Access Policy 28 (IMR11_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR11 Read Access Policy 27 (IMR11_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR11 Read Access Policy 26 (IMR11_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR11 Read Access Policy 25 (IMR11_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR11 Read Access Policy 24 (IMR11_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR11 Read Access Policy 23 (IMR11_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR11 Read Access Policy 22 (IMR11_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR11 Read Access Policy 21 (IMR11_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR11 Read Access Policy 20 (IMR11_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR11 Read Access Policy 19 (IMR11_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR11 Read Access Policy 18 (IMR11_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR11 Read Access Policy 17 (IMR11_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR11 Read Access Policy 16 (IMR11_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR11 Read Access Policy 15 (IMR11_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR11 Read Access Policy 14 (IMR11_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR11 Read Access Policy 13 (IMR11_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR11 Read Access Policy 12 (IMR11_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR11 Read Access Policy 11 (IMR11_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR11 Read Access Policy 10 (IMR11_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR11 Read Access Policy 9 (IMR11_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR11 Read Access Policy 8 (IMR11_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR11 Read Access Policy 7 (IMR11_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR11 Read Access Policy 6 (IMR11_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR11 Read Access Policy 5 (IMR11_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR11 Read Access Policy 4 (IMR11_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR11 Read Access Policy 3 (IMR11_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR11 Read Access Policy 2 (IMR11_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR11 Read Access Policy 1 (IMR11_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR11 Read Access Policy 0 (IMR11_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

## 44.3.81 IMR11 Write Access Policy (B_CR_BIMR11WAC_0_0_0_MCHBAR)—Offset 69E8h

This register, along with IMR11BASE, IMR11MASK and IMR11RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR11. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69E8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR11 Write Access Policy 63 (IMR11_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR11 Write Access Policy 62 (IMR11_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR11 Write Access Policy 61 (IMR11_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR11 Write Access Policy 60 (IMR11_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR11 Write Access Policy 59 (IMR11_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR11 Write Access Policy 58 (IMR11_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR11 Write Access Policy 57 (IMR11_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR11 Write Access Policy 56 (IMR11_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR11 Write Access Policy 55 (IMR11_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR11 Write Access Policy 54 (IMR11_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR11 Write Access Policy 53 (IMR11_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR11 Write Access Policy 52 (IMR11_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR11 Write Access Policy 51 (IMR11_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR11 Write Access Policy 50 (IMR11_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR11 Write Access Policy 49 (IMR11_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR11 Write Access Policy 48 (IMR11_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR11 Write Access Policy 47 (IMR11_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR11 Write Access Policy 46 (IMR11_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR11 Write Access Policy 45 (IMR11_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR11 Write Access Policy 44 (IMR11_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR11 Write Access Policy 43 (IMR11_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR11 Write Access Policy 42 (IMR11_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR11 Write Access Policy 41 (IMR11_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR11 Write Access Policy 40 (IMR11_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR11 Write Access Policy 39 (IMR11_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR11 Write Access Policy 38 (IMR11_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR11 Write Access Policy 37 (IMR11_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR11 Write Access Policy 36 (IMR11_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR11 Write Access Policy 35 (IMR11_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR11 Write Access Policy 34 (IMR11_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR11 Write Access Policy 33 (IMR11_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR11 Write Access Policy 32 (IMR11_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR11 Write Access Policy 31 (IMR11_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR11 Write Access Policy 30 (IMR11_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR11 Write Access Policy 29 (IMR11_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR11 Write Access Policy 28 (IMR11_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR11 Write Access Policy 27 (IMR11_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR11 Write Access Policy 26 (IMR11_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR11 Write Access Policy 25 (IMR11_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR11 Write Access Policy 24 (IMR11_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR11 Write Access Policy 23 (IMR11_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR11 Write Access Policy 22 (IMR11_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR11 Write Access Policy 21 (IMR11_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR11 Write Access Policy 20 (IMR11_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR11 Write Access Policy 19 (IMR11_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR11 Write Access Policy 18 (IMR11_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR11 Write Access Policy 17 (IMR11_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR11 Write Access Policy 16 (IMR11_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR11 Write Access Policy 15 (IMR11_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR11 Write Access Policy 14 (IMR11_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR11 Write Access Policy 13 (IMR11_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR11 Write Access Policy 12 (IMR11_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR11 Write Access Policy 11 (IMR11_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR11 Write Access Policy 10 (IMR11_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR11 Write Access Policy 9 (IMR11_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR11 Write Access Policy 8 (IMR11_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR11 Write Access Policy 7 (IMR11_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR11 Write Access Policy 6 (IMR11_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR11 Write Access Policy 5 (IMR11_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR11 Write Access Policy 4 (IMR11_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR11 Write Access Policy 3 (IMR11_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR11 Write Access Policy 2 (IMR11_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR11 Write Access Policy 1 (IMR11_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR11 Write Access Policy 0 (IMR11_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value. |

## 44.3.82 IMR12 Base (B_CR_BIMR12BASE_0_0_0_MCHBAR)—Offset 69F0h

This register, along with IMR12MASK, IMR12RAC and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR12RAC and IMR12WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69F0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR12 Base (IMR12_BASE):** Specifies bits 38:10 of the start address of IMR12 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR12MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR12 defined region. |

## 44.3.83 IMR12 Mask (B_CR_BIMR12MASK_0_0_0_MCHBAR)—Offset 69F4h

This register, along with IMR12BASE, IMR12RAC and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR12RAC and IMR12WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 69F4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR12 Mask (IMR12_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR12BASE[28:0] value. A match indicates that the incoming address falls within the IMR12 region. |

## 44.3.84 IMR12 Control Policy (B_CR_BIMR12CP_0_0_0_MCHBAR)—Offset 69F8h

This register controls the access policy to the Read Access Policy BIMR12RAC, the Write Access Policy BIMR12WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 69F8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR12 Control Policy (IMR12_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.85 IMR12 Read Access Policy (B_CR_BIMR12RAC_0_0_0_MCHBAR)—Offset 6A00h

This register, along with IMR12BASE, IMR12MASK and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR12. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A00h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR12 Read Access Policy 63 (IMR12_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR12 Read Access Policy 62 (IMR12_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR12 Read Access Policy 61 (IMR12_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR12 Read Access Policy 60 (IMR12_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR12 Read Access Policy 59 (IMR12_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR12 Read Access Policy 58 (IMR12_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR12 Read Access Policy 57 (IMR12_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR12 Read Access Policy 56 (IMR12_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR12 Read Access Policy 55 (IMR12_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR12 Read Access Policy 54 (IMR12_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR12 Read Access Policy 53 (IMR12_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR12 Read Access Policy 52 (IMR12_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR12 Read Access Policy 51 (IMR12_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR12 Read Access Policy 50 (IMR12_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR12 Read Access Policy 49 (IMR12_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR12 Read Access Policy 48 (IMR12_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR12 Read Access Policy 47 (IMR12_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR12 Read Access Policy 46 (IMR12_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR12 Read Access Policy 45 (IMR12_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR12 Read Access Policy 44 (IMR12_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR12 Read Access Policy 43 (IMR12_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR12 Read Access Policy 42 (IMR12_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR12 Read Access Policy 41 (IMR12_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR12 Read Access Policy 40 (IMR12_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR12 Read Access Policy 39 (IMR12_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR12 Read Access Policy 38 (IMR12_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR12 Read Access Policy 37 (IMR12_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR12 Read Access Policy 36 (IMR12_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR12 Read Access Policy 35 (IMR12_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR12 Read Access Policy 34 (IMR12_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR12 Read Access Policy 33 (IMR12_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR12 Read Access Policy 32 (IMR12_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR12 Read Access Policy 31 (IMR12_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR12 Read Access Policy 30 (IMR12_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR12 Read Access Policy 29 (IMR12_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR12 Read Access Policy 28 (IMR12_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR12 Read Access Policy 27 (IMR12_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR12 Read Access Policy 26 (IMR12_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR12 Read Access Policy 25 (IMR12_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR12 Read Access Policy 24 (IMR12_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR12 Read Access Policy 23 (IMR12_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR12 Read Access Policy 22 (IMR12_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR12 Read Access Policy 21 (IMR12_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR12 Read Access Policy 20 (IMR12_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR12 Read Access Policy 19 (IMR12_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR12 Read Access Policy 18 (IMR12_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR12 Read Access Policy 17 (IMR12_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR12 Read Access Policy 16 (IMR12_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR12 Read Access Policy 15 (IMR12_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR12 Read Access Policy 14 (IMR12_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR12 Read Access Policy 13 (IMR12_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR12 Read Access Policy 12 (IMR12_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR12 Read Access Policy 11 (IMR12_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR12 Read Access Policy 10 (IMR12_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR12 Read Access Policy 9 (IMR12_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR12 Read Access Policy 8 (IMR12_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR12 Read Access Policy 7 (IMR12_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR12 Read Access Policy 6 (IMR12_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR12 Read Access Policy 5 (IMR12_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR12 Read Access Policy 4 (IMR12_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR12 Read Access Policy 3 (IMR12_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR12 Read Access Policy 2 (IMR12_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR12 Read Access Policy 1 (IMR12_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR12 Read Access Policy 0 (IMR12_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

## 44.3.86 IMR12 Write Access Policy (B_CR_BIMR12WAC_0_0_0_MCHBAR)—Offset 6A08h

This register, along with IMR12BASE, IMR12MASK and IMR12RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR12. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A08h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR12 Write Access Policy 63 (IMR12_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR12 Write Access Policy 62 (IMR12_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR12 Write Access Policy 61 (IMR12_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR12 Write Access Policy 60 (IMR12_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR12 Write Access Policy 59 (IMR12_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR12 Write Access Policy 58 (IMR12_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR12 Write Access Policy 57 (IMR12_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR12 Write Access Policy 56 (IMR12_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR12 Write Access Policy 55 (IMR12_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR12 Write Access Policy 54 (IMR12_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR12 Write Access Policy 53 (IMR12_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR12 Write Access Policy 52 (IMR12_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR12 Write Access Policy 51 (IMR12_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR12 Write Access Policy 50 (IMR12_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR12 Write Access Policy 49 (IMR12_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR12 Write Access Policy 48 (IMR12_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR12 Write Access Policy 47 (IMR12_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR12 Write Access Policy 46 (IMR12_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR12 Write Access Policy 45 (IMR12_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR12 Write Access Policy 44 (IMR12_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR12 Write Access Policy 43 (IMR12_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR12 Write Access Policy 42 (IMR12_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR12 Write Access Policy 41 (IMR12_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR12 Write Access Policy 40 (IMR12_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR12 Write Access Policy 39 (IMR12_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR12 Write Access Policy 38 (IMR12_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR12 Write Access Policy 37 (IMR12_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR12 Write Access Policy 36 (IMR12_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR12 Write Access Policy 35 (IMR12_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR12 Write Access Policy 34 (IMR12_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR12 Write Access Policy 33 (IMR12_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR12 Write Access Policy 32 (IMR12_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR12 Write Access Policy 31 (IMR12_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR12 Write Access Policy 30 (IMR12_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR12 Write Access Policy 29 (IMR12_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR12 Write Access Policy 28 (IMR12_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR12 Write Access Policy 27 (IMR12_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR12 Write Access Policy 26 (IMR12_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR12 Write Access Policy 25 (IMR12_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR12 Write Access Policy 24 (IMR12_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR12 Write Access Policy 23 (IMR12_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR12 Write Access Policy 22 (IMR12_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR12 Write Access Policy 21 (IMR12_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR12 Write Access Policy 20 (IMR12_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR12 Write Access Policy 19 (IMR12_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR12 Write Access Policy 18 (IMR12_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR12 Write Access Policy 17 (IMR12_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR12 Write Access Policy 16 (IMR12_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR12 Write Access Policy 15 (IMR12_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR12 Write Access Policy 14 (IMR12_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR12 Write Access Policy 13 (IMR12_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR12 Write Access Policy 12 (IMR12_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR12 Write Access Policy 11 (IMR12_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR12 Write Access Policy 10 (IMR12_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR12 Write Access Policy 9 (IMR12_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR12 Write Access Policy 8 (IMR12_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR12 Write Access Policy 7 (IMR12_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR12 Write Access Policy 6 (IMR12_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR12 Write Access Policy 5 (IMR12_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR12 Write Access Policy 4 (IMR12_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR12 Write Access Policy 3 (IMR12_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR12 Write Access Policy 2 (IMR12_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR12 Write Access Policy 1 (IMR12_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR12 Write Access Policy 0 (IMR12_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value. |

## 44.3.87 IMR13 Base (B_CR_BIMR13BASE_0_0_0_MCHBAR)—Offset 6A10h

This register, along with IMR13MASK, IMR13RAC and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR13RAC and IMR13WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A10h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR13 Base (IMR13_BASE):** Specifies bits 38:10 of the start address of IMR13 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR13MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR13 defined region. |

## 44.3.88 IMR13 Mask (B_CR_BIMR13MASK_0_0_0_MCHBAR)—Offset 6A14h

This register, along with IMR13BASE, IMR13RAC and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR13RAC and IMR13WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A14h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR13 Mask (IMR13_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR13BASE[28:0] value. A match indicates that the incoming address falls within the IMR13 region. |

## 44.3.89 IMR13 Control Policy (B_CR_BIMR13CP_0_0_0_MCHBAR)—Offset 6A18h

This register controls the access policy to the Read Access Policy BIMR13RAC, the Write Access Policy BIMR13WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A18h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR13 Control Policy (IMR13_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.90 IMR13 Read Access Policy (B_CR_BIMR13RAC_0_0_0_MCHBAR)—Offset 6A20h

This register, along with IMR13BASE, IMR13MASK and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR13. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A20h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR13 Read Access Policy 63 (IMR13_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR13 Read Access Policy 62 (IMR13_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR13 Read Access Policy 61 (IMR13_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR13 Read Access Policy 60 (IMR13_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR13 Read Access Policy 59 (IMR13_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR13 Read Access Policy 58 (IMR13_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR13 Read Access Policy 57 (IMR13_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR13 Read Access Policy 56 (IMR13_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR13 Read Access Policy 55 (IMR13_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR13 Read Access Policy 54 (IMR13_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR13 Read Access Policy 53 (IMR13_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR13 Read Access Policy 52 (IMR13_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR13 Read Access Policy 51 (IMR13_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR13 Read Access Policy 50 (IMR13_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR13 Read Access Policy 49 (IMR13_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR13 Read Access Policy 48 (IMR13_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR13 Read Access Policy 47 (IMR13_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR13 Read Access Policy 46 (IMR13_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR13 Read Access Policy 45 (IMR13_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR13 Read Access Policy 44 (IMR13_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR13 Read Access Policy 43 (IMR13_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR13 Read Access Policy 42 (IMR13_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR13 Read Access Policy 41 (IMR13_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR13 Read Access Policy 40 (IMR13_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR13 Read Access Policy 39 (IMR13_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR13 Read Access Policy 38 (IMR13_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR13 Read Access Policy 37 (IMR13_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR13 Read Access Policy 36 (IMR13_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR13 Read Access Policy 35 (IMR13_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR13 Read Access Policy 34 (IMR13_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR13 Read Access Policy 33 (IMR13_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR13 Read Access Policy 32 (IMR13_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR13 Read Access Policy 31 (IMR13_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR13 Read Access Policy 30 (IMR13_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR13 Read Access Policy 29 (IMR13_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR13 Read Access Policy 28 (IMR13_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR13 Read Access Policy 27 (IMR13_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR13 Read Access Policy 26 (IMR13_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR13 Read Access Policy 25 (IMR13_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR13 Read Access Policy 24 (IMR13_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR13 Read Access Policy 23 (IMR13_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR13 Read Access Policy 22 (IMR13_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR13 Read Access Policy 21 (IMR13_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR13 Read Access Policy 20 (IMR13_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR13 Read Access Policy 19 (IMR13_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR13 Read Access Policy 18 (IMR13_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR13 Read Access Policy 17 (IMR13_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR13 Read Access Policy 16 (IMR13_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR13 Read Access Policy 15 (IMR13_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR13 Read Access Policy 14 (IMR13_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR13 Read Access Policy 13 (IMR13_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR13 Read Access Policy 12 (IMR13_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR13 Read Access Policy 11 (IMR13_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR13 Read Access Policy 10 (IMR13_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR13 Read Access Policy 9 (IMR13_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR13 Read Access Policy 8 (IMR13_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR13 Read Access Policy 7 (IMR13_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR13 Read Access Policy 6 (IMR13_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR13 Read Access Policy 5 (IMR13_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR13 Read Access Policy 4 (IMR13_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR13 Read Access Policy 3 (IMR13_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR13 Read Access Policy 2 (IMR13_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR13 Read Access Policy 1 (IMR13_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR13 Read Access Policy 0 (IMR13_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

## 44.3.91 IMR13 Write Access Policy (B_CR_BIMR13WAC_0_0_0_MCHBAR)—Offset 6A28h

This register, along with IMR13BASE, IMR13MASK and IMR13RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR13. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A28h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR13 Write Access Policy 63 (IMR13_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR13 Write Access Policy 62 (IMR13_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR13 Write Access Policy 61 (IMR13_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR13 Write Access Policy 60 (IMR13_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR13 Write Access Policy 59 (IMR13_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR13 Write Access Policy 58 (IMR13_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR13 Write Access Policy 57 (IMR13_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR13 Write Access Policy 56 (IMR13_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR13 Write Access Policy 55 (IMR13_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR13 Write Access Policy 54 (IMR13_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR13 Write Access Policy 53 (IMR13_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR13 Write Access Policy 52 (IMR13_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR13 Write Access Policy 51 (IMR13_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR13 Write Access Policy 50 (IMR13_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR13 Write Access Policy 49 (IMR13_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR13 Write Access Policy 48 (IMR13_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR13 Write Access Policy 47 (IMR13_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR13 Write Access Policy 46 (IMR13_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR13 Write Access Policy 45 (IMR13_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR13 Write Access Policy 44 (IMR13_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR13 Write Access Policy 43 (IMR13_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR13 Write Access Policy 42 (IMR13_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR13 Write Access Policy 41 (IMR13_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR13 Write Access Policy 40 (IMR13_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR13 Write Access Policy 39 (IMR13_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR13 Write Access Policy 38 (IMR13_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR13 Write Access Policy 37 (IMR13_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR13 Write Access Policy 36 (IMR13_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR13 Write Access Policy 35 (IMR13_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR13 Write Access Policy 34 (IMR13_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR13 Write Access Policy 33 (IMR13_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR13 Write Access Policy 32 (IMR13_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR13 Write Access Policy 31 (IMR13_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR13 Write Access Policy 30 (IMR13_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR13 Write Access Policy 29 (IMR13_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR13 Write Access Policy 28 (IMR13_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR13 Write Access Policy 27 (IMR13_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR13 Write Access Policy 26 (IMR13_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR13 Write Access Policy 25 (IMR13_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR13 Write Access Policy 24 (IMR13_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR13 Write Access Policy 23 (IMR13_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR13 Write Access Policy 22 (IMR13_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR13 Write Access Policy 21 (IMR13_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR13 Write Access Policy 20 (IMR13_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR13 Write Access Policy 19 (IMR13_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR13 Write Access Policy 18 (IMR13_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR13 Write Access Policy 17 (IMR13_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR13 Write Access Policy 16 (IMR13_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR13 Write Access Policy 15 (IMR13_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR13 Write Access Policy 14 (IMR13_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR13 Write Access Policy 13 (IMR13_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR13 Write Access Policy 12 (IMR13_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR13 Write Access Policy 11 (IMR13_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR13 Write Access Policy 10 (IMR13_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR13 Write Access Policy 9 (IMR13_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR13 Write Access Policy 8 (IMR13_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR13 Write Access Policy 7 (IMR13_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR13 Write Access Policy 6 (IMR13_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR13 Write Access Policy 5 (IMR13_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR13 Write Access Policy 4 (IMR13_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR13 Write Access Policy 3 (IMR13_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR13 Write Access Policy 2 (IMR13_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR13 Write Access Policy 1 (IMR13_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR13 Write Access Policy 0 (IMR13_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value. |

## 44.3.92 IMR14 Base (B_CR_BIMR14BASE_0_0_0_MCHBAR)—Offset 6A30h

This register, along with IMR14MASK, IMR14RAC and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR14RAC and IMR14WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A30h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR14 Base (IMR14_BASE):** Specifies bits 38:10 of the start address of IMR14 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR14MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR14 defined region. |

## 44.3.93 IMR14 Mask (B_CR_BIMR14MASK_0_0_0_MCHBAR)—Offset 6A34h

This register, along with IMR14BASE, IMR14RAC and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR14RAC and IMR14WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A34h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR14 Mask (IMR14_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR14BASE[28:0] value. A match indicates that the incoming address falls within the IMR14 region. |

## 44.3.94 IMR14 Control Policy (B_CR_BIMR14CP_0_0_0_MCHBAR)—Offset 6A38h

This register controls the access policy to the Read Access Policy BIMR14RAC, the Write Access Policy BIMR14WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A38h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR14 Control Policy (IMR14_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.95 IMR14 Read Access Policy (B_CR_BIMR14RAC_0_0_0_MCHBAR)—Offset 6A40h

This register, along with IMR14BASE, IMR14MASK and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR14. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A40h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR14 Read Access Policy 63 (IMR14_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR14 Read Access Policy 62 (IMR14_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR14 Read Access Policy 61 (IMR14_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR14 Read Access Policy 60 (IMR14_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR14 Read Access Policy 59 (IMR14_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR14 Read Access Policy 58 (IMR14_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR14 Read Access Policy 57 (IMR14_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR14 Read Access Policy 56 (IMR14_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR14 Read Access Policy 55 (IMR14_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR14 Read Access Policy 54 (IMR14_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR14 Read Access Policy 53 (IMR14_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR14 Read Access Policy 52 (IMR14_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR14 Read Access Policy 51 (IMR14_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR14 Read Access Policy 50 (IMR14_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR14 Read Access Policy 49 (IMR14_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR14 Read Access Policy 48 (IMR14_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR14 Read Access Policy 47 (IMR14_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR14 Read Access Policy 46 (IMR14_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR14 Read Access Policy 45 (IMR14_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR14 Read Access Policy 44 (IMR14_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR14 Read Access Policy 43 (IMR14_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR14 Read Access Policy 42 (IMR14_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR14 Read Access Policy 41 (IMR14_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR14 Read Access Policy 40 (IMR14_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR14 Read Access Policy 39 (IMR14_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR14 Read Access Policy 38 (IMR14_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR14 Read Access Policy 37 (IMR14_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR14 Read Access Policy 36 (IMR14_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR14 Read Access Policy 35 (IMR14_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR14 Read Access Policy 34 (IMR14_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR14 Read Access Policy 33 (IMR14_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR14 Read Access Policy 32 (IMR14_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR14 Read Access Policy 31 (IMR14_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR14 Read Access Policy 30 (IMR14_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR14 Read Access Policy 29 (IMR14_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR14 Read Access Policy 28 (IMR14_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR14 Read Access Policy 27 (IMR14_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR14 Read Access Policy 26 (IMR14_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR14 Read Access Policy 25 (IMR14_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR14 Read Access Policy 24 (IMR14_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR14 Read Access Policy 23 (IMR14_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR14 Read Access Policy 22 (IMR14_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR14 Read Access Policy 21 (IMR14_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR14 Read Access Policy 20 (IMR14_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR14 Read Access Policy 19 (IMR14_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR14 Read Access Policy 18 (IMR14_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR14 Read Access Policy 17 (IMR14_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR14 Read Access Policy 16 (IMR14_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR14 Read Access Policy 15 (IMR14_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR14 Read Access Policy 14 (IMR14_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR14 Read Access Policy 13 (IMR14_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR14 Read Access Policy 12 (IMR14_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR14 Read Access Policy 11 (IMR14_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR14 Read Access Policy 10 (IMR14_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR14 Read Access Policy 9 (IMR14_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR14 Read Access Policy 8 (IMR14_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR14 Read Access Policy 7 (IMR14_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR14 Read Access Policy 6 (IMR14_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR14 Read Access Policy 5 (IMR14_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR14 Read Access Policy 4 (IMR14_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR14 Read Access Policy 3 (IMR14_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR14 Read Access Policy 2 (IMR14_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR14 Read Access Policy 1 (IMR14_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR14 Read Access Policy 0 (IMR14_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

## 44.3.96 IMR14 Write Access Policy (B_CR_BIMR14WAC_0_0_0_MCHBAR)—Offset 6A48h

This register, along with IMR14BASE IMR14MASK and IMR14RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR14. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A48h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR14 Write Access Policy 63 (IMR14_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR14 Write Access Policy 62 (IMR14_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR14 Write Access Policy 61 (IMR14_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR14 Write Access Policy 60 (IMR14_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR14 Write Access Policy 59 (IMR14_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR14 Write Access Policy 58 (IMR14_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR14 Write Access Policy 57 (IMR14_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR14 Write Access Policy 56 (IMR14_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR14 Write Access Policy 55 (IMR14_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR14 Write Access Policy 54 (IMR14_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR14 Write Access Policy 53 (IMR14_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR14 Write Access Policy 52 (IMR14_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR14 Write Access Policy 51 (IMR14_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR14 Write Access Policy 50 (IMR14_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR14 Write Access Policy 49 (IMR14_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR14 Write Access Policy 48 (IMR14_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR14 Write Access Policy 47 (IMR14_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR14 Write Access Policy 46 (IMR14_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR14 Write Access Policy 45 (IMR14_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR14 Write Access Policy 44 (IMR14_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR14 Write Access Policy 43 (IMR14_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR14 Write Access Policy 42 (IMR14_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR14 Write Access Policy 41 (IMR14_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR14 Write Access Policy 40 (IMR14_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR14 Write Access Policy 39 (IMR14_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR14 Write Access Policy 38 (IMR14_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR14 Write Access Policy 37 (IMR14_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR14 Write Access Policy 36 (IMR14_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR14 Write Access Policy 35 (IMR14_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR14 Write Access Policy 34 (IMR14_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR14 Write Access Policy 33 (IMR14_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR14 Write Access Policy 32 (IMR14_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR14 Write Access Policy 31 (IMR14_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR14 Write Access Policy 30 (IMR14_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR14 Write Access Policy 29 (IMR14_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR14 Write Access Policy 28 (IMR14_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR14 Write Access Policy 27 (IMR14_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR14 Write Access Policy 26 (IMR14_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR14 Write Access Policy 25 (IMR14_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR14 Write Access Policy 24 (IMR14_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR14 Write Access Policy 23 (IMR14_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR14 Write Access Policy 22 (IMR14_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR14 Write Access Policy 21 (IMR14_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR14 Write Access Policy 20 (IMR14_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR14 Write Access Policy 19 (IMR14_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR14 Write Access Policy 18 (IMR14_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR14 Write Access Policy 17 (IMR14_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR14 Write Access Policy 16 (IMR14_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR14 Write Access Policy 15 (IMR14_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR14 Write Access Policy 14 (IMR14_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR14 Write Access Policy 13 (IMR14_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR14 Write Access Policy 12 (IMR14_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR14 Write Access Policy 11 (IMR14_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR14 Write Access Policy 10 (IMR14_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR14 Write Access Policy 9 (IMR14_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR14 Write Access Policy 8 (IMR14_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR14 Write Access Policy 7 (IMR14_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR14 Write Access Policy 6 (IMR14_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR14 Write Access Policy 5 (IMR14_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR14 Write Access Policy 4 (IMR14_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR14 Write Access Policy 3 (IMR14_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR14 Write Access Policy 2 (IMR14_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR14 Write Access Policy 1 (IMR14_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR14 Write Access Policy 0 (IMR14_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value. |

## 44.3.97 IMR15 Base (B_CR_BIMR15BASE_0_0_0_MCHBAR)—Offset 6A50h

This register, along with IMR15MASK, IMR15RAC and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR15RAC and IMR15WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A50h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR15 Base (IMR15_BASE):** Specifies bits 38:10 of the start address of IMR15 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR15MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR15 defined region. |

## 44.3.98 IMR15 Mask (B_CR_BIMR15MASK_0_0_0_MCHBAR)—Offset 6A54h

This register, along with IMR15BASE, IMR15RAC and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR15RAC and IMR15WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A54h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor, depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR15 Mask (IMR15_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR15BASE[28:0] value. A match indicates that the incoming address falls within the IMR15 region. |

## 44.3.99 IMR15 Control Policy (B_CR_BIMR15CP_0_0_0_MCHBAR)—Offset 6A58h

This register controls the access policy to the Read Access Policy BIMR15RAC, the Write Access Policy BIMR15WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A58h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR15 Control Policy (IMR15_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.100 IMR15 Read Access Policy (B_CR_BIMR15RAC_0_0_0_MCHBAR)—Offset 6A60h

This register, along with IMR15BASE, IMR15MASK and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR15. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A60h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR15 Read Access Policy 63 (IMR15_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR15 Read Access Policy 62 (IMR15_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR15 Read Access Policy 61 (IMR15_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR15 Read Access Policy 60 (IMR15_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR15 Read Access Policy 59 (IMR15_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR15 Read Access Policy 58 (IMR15_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR15 Read Access Policy 57 (IMR15_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR15 Read Access Policy 56 (IMR15_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR15 Read Access Policy 55 (IMR15_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR15 Read Access Policy 54 (IMR15_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR15 Read Access Policy 53 (IMR15_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR15 Read Access Policy 52 (IMR15_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR15 Read Access Policy 51 (IMR15_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR15 Read Access Policy 50 (IMR15_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR15 Read Access Policy 49 (IMR15_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR15 Read Access Policy 48 (IMR15_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR15 Read Access Policy 47 (IMR15_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR15 Read Access Policy 46 (IMR15_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR15 Read Access Policy 45 (IMR15_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR15 Read Access Policy 44 (IMR15_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR15 Read Access Policy 43 (IMR15_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR15 Read Access Policy 42 (IMR15_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR15 Read Access Policy 41 (IMR15_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR15 Read Access Policy 40 (IMR15_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR15 Read Access Policy 39 (IMR15_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR15 Read Access Policy 38 (IMR15_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR15 Read Access Policy 37 (IMR15_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR15 Read Access Policy 36 (IMR15_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR15 Read Access Policy 35 (IMR15_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR15 Read Access Policy 34 (IMR15_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR15 Read Access Policy 33 (IMR15_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR15 Read Access Policy 32 (IMR15_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR15 Read Access Policy 31 (IMR15_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR15 Read Access Policy 30 (IMR15_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR15 Read Access Policy 29 (IMR15_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR15 Read Access Policy 28 (IMR15_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR15 Read Access Policy 27 (IMR15_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR15 Read Access Policy 26 (IMR15_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR15 Read Access Policy 25 (IMR15_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR15 Read Access Policy 24 (IMR15_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR15 Read Access Policy 23 (IMR15_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR15 Read Access Policy 22 (IMR15_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR15 Read Access Policy 21 (IMR15_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR15 Read Access Policy 20 (IMR15_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR15 Read Access Policy 19 (IMR15_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR15 Read Access Policy 18 (IMR15_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR15 Read Access Policy 17 (IMR15_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR15 Read Access Policy 16 (IMR15_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR15 Read Access Policy 15 (IMR15_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR15 Read Access Policy 14 (IMR15_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR15 Read Access Policy 13 (IMR15_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR15 Read Access Policy 12 (IMR15_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR15 Read Access Policy 11 (IMR15_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR15 Read Access Policy 10 (IMR15_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR15 Read Access Policy 9 (IMR15_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR15 Read Access Policy 8 (IMR15_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR15 Read Access Policy 7 (IMR15_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR15 Read Access Policy 6 (IMR15_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR15 Read Access Policy 5 (IMR15_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR15 Read Access Policy 4 (IMR15_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR15 Read Access Policy 3 (IMR15_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR15 Read Access Policy 2 (IMR15_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR15 Read Access Policy 1 (IMR15_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR15 Read Access Policy 0 (IMR15_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

## 44.3.101 IMR15 Write Access Policy (B_CR_BIMR15WAC_0_0_0_MCHBAR)—Offset 6A68h

This register, along with IMR15BASE, IMR15MASK and IMR15RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR15. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A68h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR15 Write Access Policy 63 (IMR15_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR15 Write Access Policy 62 (IMR15_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR15 Write Access Policy 61 (IMR15_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR15 Write Access Policy 60 (IMR15_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR15 Write Access Policy 59 (IMR15_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR15 Write Access Policy 58 (IMR15_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR15 Write Access Policy 57 (IMR15_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR15 Write Access Policy 56 (IMR15_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR15 Write Access Policy 55 (IMR15_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR15 Write Access Policy 54 (IMR15_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR15 Write Access Policy 53 (IMR15_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR15 Write Access Policy 52 (IMR15_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR15 Write Access Policy 51 (IMR15_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR15 Write Access Policy 50 (IMR15_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR15 Write Access Policy 49 (IMR15_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR15 Write Access Policy 48 (IMR15_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR15 Write Access Policy 47 (IMR15_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR15 Write Access Policy 46 (IMR15_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR15 Write Access Policy 45 (IMR15_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR15 Write Access Policy 44 (IMR15_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR15 Write Access Policy 43 (IMR15_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR15 Write Access Policy 42 (IMR15_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR15 Write Access Policy 41 (IMR15_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR15 Write Access Policy 40 (IMR15_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR15 Write Access Policy 39 (IMR15_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR15 Write Access Policy 38 (IMR15_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR15 Write Access Policy 37 (IMR15_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR15 Write Access Policy 36 (IMR15_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR15 Write Access Policy 35 (IMR15_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR15 Write Access Policy 34 (IMR15_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR15 Write Access Policy 33 (IMR15_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR15 Write Access Policy 32 (IMR15_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR15 Write Access Policy 31 (IMR15_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR15 Write Access Policy 30 (IMR15_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR15 Write Access Policy 29 (IMR15_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR15 Write Access Policy 28 (IMR15_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR15 Write Access Policy 27 (IMR15_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR15 Write Access Policy 26 (IMR15_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR15 Write Access Policy 25 (IMR15_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR15 Write Access Policy 24 (IMR15_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR15 Write Access Policy 23 (IMR15_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR15 Write Access Policy 22 (IMR15_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR15 Write Access Policy 21 (IMR15_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR15 Write Access Policy 20 (IMR15_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR15 Write Access Policy 19 (IMR15_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR15 Write Access Policy 18 (IMR15_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR15 Write Access Policy 17 (IMR15_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR15 Write Access Policy 16 (IMR15_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR15 Write Access Policy 15 (IMR15_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR15 Write Access Policy 14 (IMR15_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR15 Write Access Policy 13 (IMR15_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR15 Write Access Policy 12 (IMR15_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR15 Write Access Policy 11 (IMR15_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR15 Write Access Policy 10 (IMR15_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR15 Write Access Policy 9 (IMR15_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR15 Write Access Policy 8 (IMR15_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR15 Write Access Policy 7 (IMR15_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR15 Write Access Policy 6 (IMR15_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR15 Write Access Policy 5 (IMR15_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR15 Write Access Policy 4 (IMR15_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR15 Write Access Policy 3 (IMR15_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR15 Write Access Policy 2 (IMR15_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR15 Write Access Policy 1 (IMR15_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR15 Write Access Policy 0 (IMR15_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value. |

## 44.3.102 IMR16 Base (B_CR_BIMR16BASE_0_0_0_MCHBAR)—Offset 6A70h

This register, along with IMR16MASK, IMR16RAC and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR16RAC and IMR16WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A70h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR16 Base (IMR16_BASE):** Specifies bits 38:10 of the start address of IMR16 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR16MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR16 defined region. |

## 44.3.103 IMR16 Mask (B_CR_BIMR16MASK_0_0_0_MCHBAR)—Offset 6A74h

This register, along with IMR16BASE, IMR16RAC and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR16RAC and IMR16WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A74h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR16 Mask (IMR16_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR16BASE[28:0] value. A match indicates that the incoming address falls within the IMR16 region. |

## 44.3.104 IMR16 Control Policy (B_CR_BIMR16CP_0_0_0_MCHBAR)—Offset 6A78h

This register controls the access policy to the Read Access Policy BIMR16RAC, the Write Access Policy BIMR16WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A78h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR16 Control Policy (IMR16_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.105 IMR16 Read Access Policy (B_CR_BIMR16RAC_0_0_0_MCHBAR)—Offset 6A80h

This register, along with IMR16BASE, IMR16MASK and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR16. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A80h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR16 Read Access Policy 63 (IMR16_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR16 Read Access Policy 62 (IMR16_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR16 Read Access Policy 61 (IMR16_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR16 Read Access Policy 60 (IMR16_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR16 Read Access Policy 59 (IMR16_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR16 Read Access Policy 58 (IMR16_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR16 Read Access Policy 57 (IMR16_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR16 Read Access Policy 56 (IMR16_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR16 Read Access Policy 55 (IMR16_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR16 Read Access Policy 54 (IMR16_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR16 Read Access Policy 53 (IMR16_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR16 Read Access Policy 52 (IMR16_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR16 Read Access Policy 51 (IMR16_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR16 Read Access Policy 50 (IMR16_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR16 Read Access Policy 49 (IMR16_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR16 Read Access Policy 48 (IMR16_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR16 Read Access Policy 47 (IMR16_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR16 Read Access Policy 46 (IMR16_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR16 Read Access Policy 45 (IMR16_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR16 Read Access Policy 44 (IMR16_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR16 Read Access Policy 43 (IMR16_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR16 Read Access Policy 42 (IMR16_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR16 Read Access Policy 41 (IMR16_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR16 Read Access Policy 40 (IMR16_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR16 Read Access Policy 39 (IMR16_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR16 Read Access Policy 38 (IMR16_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR16 Read Access Policy 37 (IMR16_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR16 Read Access Policy 36 (IMR16_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR16 Read Access Policy 35 (IMR16_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR16 Read Access Policy 34 (IMR16_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR16 Read Access Policy 33 (IMR16_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR16 Read Access Policy 32 (IMR16_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR16 Read Access Policy 31 (IMR16_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR16 Read Access Policy 30 (IMR16_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR16 Read Access Policy 29 (IMR16_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR16 Read Access Policy 28 (IMR16_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR16 Read Access Policy 27 (IMR16_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR16 Read Access Policy 26 (IMR16_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR16 Read Access Policy 25 (IMR16_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR16 Read Access Policy 24 (IMR16_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR16 Read Access Policy 23 (IMR16_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR16 Read Access Policy 22 (IMR16_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR16 Read Access Policy 21 (IMR16_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR16 Read Access Policy 20 (IMR16_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR16 Read Access Policy 19 (IMR16_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR16 Read Access Policy 18 (IMR16_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR16 Read Access Policy 17 (IMR16_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR16 Read Access Policy 16 (IMR16_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR16 Read Access Policy 15 (IMR16_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR16 Read Access Policy 14 (IMR16_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR16 Read Access Policy 13 (IMR16_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR16 Read Access Policy 12 (IMR16_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR16 Read Access Policy 11 (IMR16_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR16 Read Access Policy 10 (IMR16_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR16 Read Access Policy 9 (IMR16_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR16 Read Access Policy 8 (IMR16_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR16 Read Access Policy 7 (IMR16_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR16 Read Access Policy 6 (IMR16_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR16 Read Access Policy 5 (IMR16_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR16 Read Access Policy 4 (IMR16_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR16 Read Access Policy 3 (IMR16_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR16 Read Access Policy 2 (IMR16_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR16 Read Access Policy 1 (IMR16_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR16 Read Access Policy 0 (IMR16_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

## 44.3.106 IMR16 Write Access Policy (B_CR_BIMR16WAC_0_0_0_MCHBAR)—Offset 6A88h

This register, along with IMR16BASE, IMR16MASK and IMR16RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR16. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A88h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR16 Write Access Policy 63 (IMR16_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR16 Write Access Policy 62 (IMR16_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR16 Write Access Policy 61 (IMR16_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR16 Write Access Policy 60 (IMR16_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR16 Write Access Policy 59 (IMR16_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR16 Write Access Policy 58 (IMR16_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR16 Write Access Policy 57 (IMR16_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR16 Write Access Policy 56 (IMR16_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR16 Write Access Policy 55 (IMR16_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR16 Write Access Policy 54 (IMR16_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR16 Write Access Policy 53 (IMR16_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR16 Write Access Policy 52 (IMR16_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR16 Write Access Policy 51 (IMR16_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR16 Write Access Policy 50 (IMR16_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR16 Write Access Policy 49 (IMR16_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR16 Write Access Policy 48 (IMR16_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR16 Write Access Policy 47 (IMR16_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR16 Write Access Policy 46 (IMR16_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR16 Write Access Policy 45 (IMR16_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR16 Write Access Policy 44 (IMR16_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR16 Write Access Policy 43 (IMR16_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR16 Write Access Policy 42 (IMR16_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR16 Write Access Policy 41 (IMR16_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR16 Write Access Policy 40 (IMR16_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR16 Write Access Policy 39 (IMR16_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR16 Write Access Policy 38 (IMR16_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR16 Write Access Policy 37 (IMR16_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR16 Write Access Policy 36 (IMR16_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR16 Write Access Policy 35 (IMR16_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR16 Write Access Policy 34 (IMR16_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR16 Write Access Policy 33 (IMR16_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR16 Write Access Policy 32 (IMR16_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR16 Write Access Policy 31 (IMR16_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR16 Write Access Policy 30 (IMR16_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR16 Write Access Policy 29 (IMR16_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR16 Write Access Policy 28 (IMR16_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR16 Write Access Policy 27 (IMR16_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR16 Write Access Policy 26 (IMR16_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR16 Write Access Policy 25 (IMR16_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR16 Write Access Policy 24 (IMR16_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR16 Write Access Policy 23 (IMR16_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR16 Write Access Policy 22 (IMR16_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR16 Write Access Policy 21 (IMR16_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR16 Write Access Policy 20 (IMR16_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR16 Write Access Policy 19 (IMR16_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR16 Write Access Policy 18 (IMR16_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR16 Write Access Policy 17 (IMR16_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR16 Write Access Policy 16 (IMR16_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR16 Write Access Policy 15 (IMR16_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR16 Write Access Policy 14 (IMR16_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR16 Write Access Policy 13 (IMR16_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR16 Write Access Policy 12 (IMR16_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR16 Write Access Policy 11 (IMR16_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR16 Write Access Policy 10 (IMR16_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR16 Write Access Policy 9 (IMR16_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR16 Write Access Policy 8 (IMR16_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR16 Write Access Policy 7 (IMR16_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR16 Write Access Policy 6 (IMR16_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR16 Write Access Policy 5 (IMR16_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR16 Write Access Policy 4 (IMR16_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR16 Write Access Policy 3 (IMR16_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR16 Write Access Policy 2 (IMR16_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR16 Write Access Policy 1 (IMR16_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR16 Write Access Policy 0 (IMR16_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value. |

## 44.3.107 IMR17 Base (B_CR_BIMR17BASE_0_0_0_MCHBAR)—Offset 6A90h

This register, along with IMR17MASK, IMR17RAC and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR17RAC and IMR17WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A90h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR17BASE (IMR17_BASE):** Specifies bits 38:10 of the start address of IMR17 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR17MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR17 defined region. |

## 44.3.108 IMR17 Mask (B_CR_BIMR17MASK_0_0_0_MCHBAR)—Offset 6A94h

This register, along with IMR17BASE, IMR17RAC and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR17RAC and IMR17WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6A94h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor, depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR17 Mask (IMR17_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR17BASE[28:0] value. A match indicates that the incoming address falls within the IMR17 region. |

## 44.3.109 IMR17 Control Policy (B_CR_BIMR17CP_0_0_0_MCHBAR)—Offset 6A98h

This register controls the access policy to the Read Access Policy BIMR17RAC, the Write Access Policy BIMR17WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6A98h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR17 Control Policy (IMR17_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.110 IMR17 Read Access Policy (B_CR_BIMR17RAC_0_0_0_MCHBAR)—Offset 6AA0h

This register, along with IMR17BASE, IMR17MASK and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR17. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AA0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR17 Read Access Policy 63 (IMR17_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR17 Read Access Policy 62 (IMR17_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR17 Read Access Policy 61 (IMR17_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR17 Read Access Policy 60 (IMR17_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR17 Read Access Policy 59 (IMR17_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR17 Read Access Policy 58 (IMR17_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR17 Read Access Policy 57 (IMR17_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR17 Read Access Policy 56 (IMR17_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR17 Read Access Policy 55 (IMR17_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR17 Read Access Policy 54 (IMR17_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR17 Read Access Policy 53 (IMR17_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR17 Read Access Policy 52 (IMR17_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR17 Read Access Policy 51 (IMR17_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR17 Read Access Policy 50 (IMR17_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR17 Read Access Policy 49 (IMR17_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR17 Read Access Policy 48 (IMR17_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR17 Read Access Policy 47 (IMR17_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR17 Read Access Policy 46 (IMR17_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR17 Read Access Policy 45 (IMR17_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR17 Read Access Policy 44 (IMR17_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR17 Read Access Policy 43 (IMR17_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR17 Read Access Policy 42 (IMR17_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR17 Read Access Policy 41 (IMR17_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR17 Read Access Policy 40 (IMR17_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR17 Read Access Policy 39 (IMR17_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR17 Read Access Policy 38 (IMR17_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR17 Read Access Policy 37 (IMR17_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR17 Read Access Policy 36 (IMR17_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR17 Read Access Policy 35 (IMR17_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR17 Read Access Policy 34 (IMR17_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR17 Read Access Policy 33 (IMR17_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR17 Read Access Policy 32 (IMR17_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR17 Read Access Policy 31 (IMR17_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR17 Read Access Policy 30 (IMR17_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR17 Read Access Policy 29 (IMR17_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR17 Read Access Policy 28 (IMR17_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR17 Read Access Policy 27 (IMR17_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR17 Read Access Policy 26 (IMR17_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR17 Read Access Policy 25 (IMR17_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR17 Read Access Policy 24 (IMR17_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR17 Read Access Policy 23 (IMR17_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR17 Read Access Policy 22 (IMR17_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR17 Read Access Policy 21 (IMR17_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR17 Read Access Policy 20 (IMR17_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR17 Read Access Policy 19 (IMR17_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR17 Read Access Policy 18 (IMR17_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR17 Read Access Policy 17 (IMR17_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR17 Read Access Policy 16 (IMR17_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR17 Read Access Policy 15 (IMR17_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR17 Read Access Policy 14 (IMR17_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR17 Read Access Policy 13 (IMR17_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR17 Read Access Policy 12 (IMR17_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR17 Read Access Policy 11 (IMR17_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR17 Read Access Policy 10 (IMR17_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR17 Read Access Policy 9 (IMR17_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR17 Read Access Policy 8 (IMR17_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR17 Read Access Policy 7 (IMR17_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR17 Read Access Policy 6 (IMR17_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR17 Read Access Policy 5 (IMR17_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR17 Read Access Policy 4 (IMR17_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR17 Read Access Policy 3 (IMR17_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR17 Read Access Policy 2 (IMR17_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR17 Read Access Policy 1 (IMR17_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR17 Read Access Policy 0 (IMR17_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

## 44.3.111 IMR17 Write Access Policy (B_CR_BIMR17WAC_0_0_0_MCHBAR)—Offset 6AA8h

This register, along with IMR17BASE, IMR17MASK and IMR17RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR17. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AA8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR17 Write Access Policy 63 (IMR17_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR17 Write Access Policy 62 (IMR17_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR17 Write Access Policy 61 (IMR17_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR17 Write Access Policy 60 (IMR17_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR17 Write Access Policy 59 (IMR17_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR17 Write Access Policy 58 (IMR17_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR17 Write Access Policy 57 (IMR17_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR17 Write Access Policy 56 (IMR17_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR17 Write Access Policy 55 (IMR17_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR17 Write Access Policy 54 (IMR17_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR17 Write Access Policy 53 (IMR17_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR17 Write Access Policy 52 (IMR17_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR17 Write Access Policy 51 (IMR17_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR17 Write Access Policy 50 (IMR17_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR17 Write Access Policy 49 (IMR17_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR17 Write Access Policy 48 (IMR17_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR17 Write Access Policy 47 (IMR17_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR17 Write Access Policy 46 (IMR17_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR17 Write Access Policy 45 (IMR17_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR17 Write Access Policy 44 (IMR17_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR17 Write Access Policy 43 (IMR17_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR17 Write Access Policy 42 (IMR17_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR17 Write Access Policy 41 (IMR17_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR17 Write Access Policy 40 (IMR17_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR17 Write Access Policy 39 (IMR17_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR17 Write Access Policy 38 (IMR17_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR17 Write Access Policy 37 (IMR17_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR17 Write Access Policy 36 (IMR17_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR17 Write Access Policy 35 (IMR17_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR17 Write Access Policy 34 (IMR17_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR17 Write Access Policy 33 (IMR17_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR17 Write Access Policy 32 (IMR17_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR17 Write Access Policy 31 (IMR17_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR17 Write Access Policy 30 (IMR17_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR17 Write Access Policy 29 (IMR17_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR17 Write Access Policy 28 (IMR17_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR17 Write Access Policy 27 (IMR17_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR17 Write Access Policy 26 (IMR17_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR17 Write Access Policy 25 (IMR17_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR17 Write Access Policy 24 (IMR17_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR17 Write Access Policy 23 (IMR17_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR17 Write Access Policy 22 (IMR17_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR17 Write Access Policy 21 (IMR17_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR17 Write Access Policy 20 (IMR17_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR17 Write Access Policy 19 (IMR17_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR17 Write Access Policy 18 (IMR17_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR17 Write Access Policy 17 (IMR17_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR17 Write Access Policy 16 (IMR17_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR17 Write Access Policy 15 (IMR17_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR17 Write Access Policy 14 (IMR17_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR17 Write Access Policy 13 (IMR17_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR17 Write Access Policy 12 (IMR17_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR17 Write Access Policy 11 (IMR17_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR17 Write Access Policy 10 (IMR17_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR17 Write Access Policy 9 (IMR17_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR17 Write Access Policy 8 (IMR17_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR17 Write Access Policy 7 (IMR17_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR17 Write Access Policy 6 (IMR17_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR17 Write Access Policy 5 (IMR17_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR17 Write Access Policy 4 (IMR17_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR17 Write Access Policy 3 (IMR17_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR17 Write Access Policy 2 (IMR17_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR17 Write Access Policy 1 (IMR17_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR17 Write Access Policy 0 (IMR17_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value. |

## 44.3.112 IMR18 Base (B_CR_BIMR18BASE_0_0_0_MCHBAR)—Offset 6AB0h

This register, along with IMR18MASK, IMR18RAC and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR18RAC and IMR18WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6AB0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR18 Base (IMR18_BASE):** Specifies bits 38:10 of the start address of IMR18 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR18MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR18 defined region. |

## 44.3.113 IMR18 Mask (B_CR_BIMR18MASK_0_0_0_MCHBAR)—Offset 6AB4h

This register, along with IMR18BASE, IMR18RAC and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR18RAC and IMR18WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6AB4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR18 Mask (IMR18_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR18BASE[28:0] value. A match indicates that the incoming address falls within the IMR18 region. |

## 44.3.114 IMR18 Control Policy (B_CR_BIMR18CP_0_0_0_MCHBAR)—Offset 6AB8h

This register controls the access policy to the Read Access Policy BIMR18RAC, the Write Access Policy BIMR18WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AB8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR18 Control Policy (IMR18_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.115 IMR18 Read Access Policy (B_CR_BIMR18RAC_0_0_0_MCHBAR)—Offset 6AC0h

This register, along with IMR18BASE, IMR18MASK and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR18. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AC0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR18 Read Access Policy 63 (IMR18_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR18 Read Access Policy 62 (IMR18_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR18 Read Access Policy 61 (IMR18_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR18 Read Access Policy 60 (IMR18_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR18 Read Access Policy 59 (IMR18_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR18 Read Access Policy 58 (IMR18_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR18 Read Access Policy 57 (IMR18_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR18 Read Access Policy 56 (IMR18_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR18 Read Access Policy 55 (IMR18_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR18 Read Access Policy 54 (IMR18_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR18 Read Access Policy 53 (IMR18_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR18 Read Access Policy 52 (IMR18_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR18 Read Access Policy 51 (IMR18_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR18 Read Access Policy 50 (IMR18_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR18 Read Access Policy 49 (IMR18_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR18 Read Access Policy 48 (IMR18_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR18 Read Access Policy 47 (IMR18_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR18 Read Access Policy 46 (IMR18_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR18 Read Access Policy 45 (IMR18_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR18 Read Access Policy 44 (IMR18_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR18 Read Access Policy 43 (IMR18_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR18 Read Access Policy 42 (IMR18_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR18 Read Access Policy 41 (IMR18_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR18 Read Access Policy 40 (IMR18_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR18 Read Access Policy 39 (IMR18_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR18 Read Access Policy 38 (IMR18_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR18 Read Access Policy 37 (IMR18_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR18 Read Access Policy 36 (IMR18_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR18 Read Access Policy 35 (IMR18_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR18 Read Access Policy 34 (IMR18_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR18 Read Access Policy 33 (IMR18_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR18 Read Access Policy 32 (IMR18_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR18 Read Access Policy 31 (IMR18_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR18 Read Access Policy 30 (IMR18_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR18 Read Access Policy 29 (IMR18_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR18 Read Access Policy 28 (IMR18_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR18 Read Access Policy 27 (IMR18_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR18 Read Access Policy 26 (IMR18_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR18 Read Access Policy 25 (IMR18_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR18 Read Access Policy 24 (IMR18_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR18 Read Access Policy 23 (IMR18_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR18 Read Access Policy 22 (IMR18_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR18 Read Access Policy 21 (IMR18_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR18 Read Access Policy 20 (IMR18_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR18 Read Access Policy 19 (IMR18_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR18 Read Access Policy 18 (IMR18_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR18 Read Access Policy 17 (IMR18_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR18 Read Access Policy 16 (IMR18_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR18 Read Access Policy 15 (IMR18_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR18 Read Access Policy 14 (IMR18_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR18 Read Access Policy 13 (IMR18_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR18 Read Access Policy 12 (IMR18_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR18 Read Access Policy 11 (IMR18_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR18 Read Access Policy 10 (IMR18_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR18 Read Access Policy 9 (IMR18_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR18 Read Access Policy 8 (IMR18_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR18 Read Access Policy 7 (IMR18_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR18 Read Access Policy 6 (IMR18_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR18 Read Access Policy 5 (IMR18_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR18 Read Access Policy 4 (IMR18_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR18 Read Access Policy 3 (IMR18_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR18 Read Access Policy 2 (IMR18_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR18 Read Access Policy 1 (IMR18_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR18 Read Access Policy 0 (IMR18_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

## 44.3.116 IMR18 Write Access Policy (B_CR_BIMR18WAC_0_0_0_MCHBAR)—Offset 6AC8h

This register, along with IMR18BASE, IMR18MASK and IMR18RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR18. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AC8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR18_WRITE_POL_63:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR18_WRITE_POL_62:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR18_WRITE_POL_61:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR18_WRITE_POL_60:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR18_WRITE_POL_59:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR18_WRITE_POL_58:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR18_WRITE_POL_57:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR18_WRITE_POL_56:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR18_WRITE_POL_55:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR18_WRITE_POL_54:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR18_WRITE_POL_53:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR18_WRITE_POL_52:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR18_WRITE_POL_51:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR18_WRITE_POL_50:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR18_WRITE_POL_49:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR18_WRITE_POL_48:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR18_WRITE_POL_47:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR18_WRITE_POL_46:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR18_WRITE_POL_45:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR18_WRITE_POL_44:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR18_WRITE_POL_43:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR18_WRITE_POL_42:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR18_WRITE_POL_41:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR18_WRITE_POL_40:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR18_WRITE_POL_39:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR18_WRITE_POL_38:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR18_WRITE_POL_37:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR18_WRITE_POL_36:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR18_WRITE_POL_35:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR18_WRITE_POL_34:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR18_WRITE_POL_33:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR18_WRITE_POL_32:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR18_WRITE_POL_31:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR18_WRITE_POL_30:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR18_WRITE_POL_29:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR18_WRITE_POL_28:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR18_WRITE_POL_27:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR18_WRITE_POL_26:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR18_WRITE_POL_25:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR18_WRITE_POL_24:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR18_WRITE_POL_23:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR18_WRITE_POL_22:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR18_WRITE_POL_21:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR18_WRITE_POL_20:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR18_WRITE_POL_19:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR18_WRITE_POL_18:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR18_WRITE_POL_17:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR18_WRITE_POL_16:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR18_WRITE_POL_15:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR18_WRITE_POL_14:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR18_WRITE_POL_13:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR18_WRITE_POL_12:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR18_WRITE_POL_11:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR18_WRITE_POL_10:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR18_WRITE_POL_9:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR18_WRITE_POL_8:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR18_WRITE_POL_7:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR18_WRITE_POL_6:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR18_WRITE_POL_5:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR18_WRITE_POL_4:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR18_WRITE_POL_3:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR18_WRITE_POL_2:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR18_WRITE_POL_1:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR18_WRITE_POL_0:** B-Unit IMR18 Write Access Policy: Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value. |

## 44.3.117 IMR19 Base (B_CR_BIMR19BASE_0_0_0_MCHBAR)—Offset 6AD0h

This register, along with IMR19MASK, IMR19RAC and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR19RAC and IMR19WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6AD0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **IMR Enable (IMR_EN):** Enables access checking for the IMR region. |
| 30 | 0x0<br>RW | **Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents. |
| 29 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 28:0 | 0x0<br>RW | **Base 0 IMR19 Base (IMR19_BASE):** Specifies bits 38:10 of the start address of IMR19 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR19MASK[28:0] applied to bits 38:10 of the incoming address to determine if an access falls within the IMR19 defined region. |

## 44.3.118 IMR19 Mask (B_CR_BIMR19MASK_0_0_0_MCHBAR)—Offset 6AD4h

This register, along with IMR19BASE, IMR19RAC and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR19RAC and IMR19WAC registers.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6AD4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requestor. When set to 0, inhibits HITM data from GT from being returned to the requestor. HITM data from IA cores may be returned to the requestor depending on the setting of the IA_IWB_EN bit. |
| 30 | 0x0<br>RW | **Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requestor. When set to 0, inhibits HITM data from IA cores from being returned to the requestor. HITM data from GT may be returned to the requestor depending on the setting of the GT_IWB_EN bit. |
| 29 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 28:0 | 0x0<br>RW | **Mask 0 IMR19 Mask (IMR19_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR19BASE[28:0] value. A match indicates that the incoming address falls within the IMR19 region. |

## 44.3.119 IMR19 Control Policy (B_CR_BIMR19CP_0_0_0_MCHBAR)—Offset 6AD8h

This register controls the access policy to the Read Access Policy BIMR19RAC, the Write Access Policy BIMR19WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AD8h

**Default:** 00000C0061010202h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0xc0061010202<br>RW | **IMR19 Control Policy (IMR19_CTRL_POL):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC, BIMR19CP registers, based on the value from each agent's 6bit SAI field. |

## 44.3.120 IMR19 Read Access Policy (B_CR_BIMR19RAC_0_0_0_MCHBAR)—Offset 6AE0h

This register, along with IMR19BASE, IMR19MASK and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR19. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AE0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR19 Read Access Policy 63 (IMR19_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR19 Read Access Policy 62 (IMR19_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR19 Read Access Policy 61 (IMR19_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR19 Read Access Policy 60 (IMR19_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR19 Read Access Policy 59 (IMR19_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR19 Read Access Policy 58 (IMR19_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR19 Read Access Policy 57 (IMR19_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR19 Read Access Policy 56 (IMR19_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR19 Read Access Policy 55 (IMR19_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR19 Read Access Policy 54 (IMR19_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR19 Read Access Policy 53 (IMR19_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR19 Read Access Policy 52 (IMR19_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR19 Read Access Policy 51 (IMR19_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR19 Read Access Policy 50 (IMR19_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR19 Read Access Policy 49 (IMR19_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR19 Read Access Policy 48 (IMR19_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR19 Read Access Policy 47 (IMR19_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR19 Read Access Policy 46 (IMR19_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR19 Read Access Policy 45 (IMR19_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR19 Read Access Policy 44 (IMR19_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR19 Read Access Policy 43 (IMR19_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR19 Read Access Policy 42 (IMR19_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR19 Read Access Policy 41 (IMR19_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR19 Read Access Policy 40 (IMR19_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR19 Read Access Policy 39 (IMR19_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR19 Read Access Policy 38 (IMR19_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR19 Read Access Policy 37 (IMR19_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR19 Read Access Policy 36 (IMR19_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR19 Read Access Policy 35 (IMR19_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR19 Read Access Policy 34 (IMR19_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR19 Read Access Policy 33 (IMR19_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR19 Read Access Policy 32 (IMR19_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR19 Read Access Policy 31 (IMR19_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR19 Read Access Policy 30 (IMR19_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR19 Read Access Policy 29 (IMR19_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR19 Read Access Policy 28 (IMR19_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR19 Read Access Policy 27 (IMR19_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR19 Read Access Policy 26 (IMR19_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR19 Read Access Policy 25 (IMR19_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR19 Read Access Policy 24 (IMR19_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR19 Read Access Policy 23 (IMR19_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR19 Read Access Policy 22 (IMR19_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR19 Read Access Policy 21 (IMR19_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR19 Read Access Policy 20 (IMR19_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR19 Read Access Policy 19 (IMR19_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR19 Read Access Policy 18 (IMR19_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR19 Read Access Policy 17 (IMR19_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR19 Read Access Policy 16 (IMR19_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR19 Read Access Policy 15 (IMR19_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR19 Read Access Policy 14 (IMR19_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR19 Read Access Policy 13 (IMR19_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR19 Read Access Policy 12 (IMR19_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR19 Read Access Policy 11 (IMR19_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR19 Read Access Policy 10 (IMR19_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR19 Read Access Policy 9 (IMR19_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR19 Read Access Policy 8 (IMR19_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR19 Read Access Policy 7 (IMR19_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR19 Read Access Policy 6 (IMR19_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR19 Read Access Policy 5 (IMR19_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR19 Read Access Policy 4 (IMR19_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR19 Read Access Policy 3 (IMR19_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR19 Read Access Policy 2 (IMR19_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR19 Read Access Policy 1 (IMR19_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR19 Read Access Policy 0 (IMR19_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

## 44.3.121 IMR19 Write Access Policy (B_CR_BIMR19WAC_0_0_0_MCHBAR)—Offset 6AE8h

This register, along with IMR19BASE, IMR19MASK and IMR19RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR19. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6AE8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW | **IMR19 Write Access Policy 63 (IMR19_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 62 | 0x0<br>RO | **IMR19 Write Access Policy 62 (IMR19_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 61 | 0x0<br>RO | **IMR19 Write Access Policy 61 (IMR19_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 60 | 0x0<br>RO | **IMR19 Write Access Policy 60 (IMR19_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 59 | 0x0<br>RO | **IMR19 Write Access Policy 59 (IMR19_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 58 | 0x0<br>RO | **IMR19 Write Access Policy 58 (IMR19_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 57 | 0x0<br>RO | **IMR19 Write Access Policy 57 (IMR19_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 56 | 0x0<br>RW | **IMR19 Write Access Policy 56 (IMR19_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 55 | 0x0<br>RW | **IMR19 Write Access Policy 55 (IMR19_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 54 | 0x0<br>RW | **IMR19 Write Access Policy 54 (IMR19_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 53 | 0x0<br>RO | **IMR19 Write Access Policy 53 (IMR19_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 52 | 0x0<br>RO | **IMR19 Write Access Policy 52 (IMR19_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 51 | 0x0<br>RO | **IMR19 Write Access Policy 51 (IMR19_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 50 | 0x0<br>RW | **IMR19 Write Access Policy 50 (IMR19_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 49 | 0x0<br>RW | **IMR19 Write Access Policy 49 (IMR19_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 48 | 0x0<br>RW | **IMR19 Write Access Policy 48 (IMR19_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 47 | 0x0<br>RO | **IMR19 Write Access Policy 47 (IMR19_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 46 | 0x0<br>RO | **IMR19 Write Access Policy 46 (IMR19_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 45 | 0x0<br>RO | **IMR19 Write Access Policy 45 (IMR19_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 44 | 0x0<br>RW | **IMR19 Write Access Policy 44 (IMR19_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 43 | 0x0<br>RW | **IMR19 Write Access Policy 43 (IMR19_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 42 | 0x0<br>RW | **IMR19 Write Access Policy 42 (IMR19_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 41 | 0x0<br>RW | **IMR19 Write Access Policy 41 (IMR19_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 40 | 0x0<br>RW | **IMR19 Write Access Policy 40 (IMR19_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 39 | 0x0<br>RO | **IMR19 Write Access Policy 39 (IMR19_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 38 | 0x0<br>RW | **IMR19 Write Access Policy 38 (IMR19_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 37 | 0x0<br>RO | **IMR19 Write Access Policy 37 (IMR19_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 36 | 0x0<br>RW | **IMR19 Write Access Policy 36 (IMR19_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 35 | 0x0<br>RO | **IMR19 Write Access Policy 35 (IMR19_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 34 | 0x0<br>RW | **IMR19 Write Access Policy 34 (IMR19_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 33 | 0x0<br>RW | **IMR19 Write Access Policy 33 (IMR19_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 32 | 0x0<br>RW | **IMR19 Write Access Policy 32 (IMR19_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 31 | 0x0<br>RO | **IMR19 Write Access Policy 31 (IMR19_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 30 | 0x0<br>RW | **IMR19 Write Access Policy 30 (IMR19_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 29 | 0x0<br>RW | **IMR19 Write Access Policy 29 (IMR19_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 28 | 0x0<br>RW | **IMR19 Write Access Policy 28 (IMR19_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 27 | 0x0<br>RW | **IMR19 Write Access Policy 27 (IMR19_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 26 | 0x0<br>RW | **IMR19 Write Access Policy 26 (IMR19_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 25 | 0x0<br>RW | **IMR19 Write Access Policy 25 (IMR19_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 24 | 0x0<br>RW | **IMR19 Write Access Policy 24 (IMR19_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 23 | 0x0<br>RO | **IMR19 Write Access Policy 23 (IMR19_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 22 | 0x0<br>RO | **IMR19 Write Access Policy 22 (IMR19_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 21 | 0x0<br>RW | **IMR19 Write Access Policy 21 (IMR19_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 20 | 0x0<br>RO | **IMR19 Write Access Policy 20 (IMR19_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 19 | 0x0<br>RO | **IMR19 Write Access Policy 19 (IMR19_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 18 | 0x0<br>RO | **IMR19 Write Access Policy 18 (IMR19_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 17 | 0x0<br>RW | **IMR19 Write Access Policy 17 (IMR19_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 16 | 0x0<br>RW | **IMR19 Write Access Policy 16 (IMR19_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 15 | 0x0<br>RO | **IMR19 Write Access Policy 15 (IMR19_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 14 | 0x0<br>RO | **IMR19 Write Access Policy 14 (IMR19_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 13 | 0x0<br>RW | **IMR19 Write Access Policy 13 (IMR19_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 12 | 0x0<br>RW | **IMR19 Write Access Policy 12 (IMR19_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 11 | 0x0<br>RO | **IMR19 Write Access Policy 11 (IMR19_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 10 | 0x0<br>RO | **IMR19 Write Access Policy 10 (IMR19_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 9 | 0x0<br>RO | **IMR19 Write Access Policy 9 (IMR19_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 8 | 0x0<br>RW | **IMR19 Write Access Policy 8 (IMR19_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 7 | 0x0<br>RO | **IMR19 Write Access Policy 7 (IMR19_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 6 | 0x0<br>RO | **IMR19 Write Access Policy 6 (IMR19_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 5 | 0x0<br>RW | **IMR19 Write Access Policy 5 (IMR19_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 4 | 0x0<br>RW | **IMR19 Write Access Policy 4 (IMR19_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 3 | 0x0<br>RW | **IMR19 Write Access Policy 3 (IMR19_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 2 | 0x0<br>RW | **IMR19 Write Access Policy 2 (IMR19_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 1 | 0x0<br>RW | **IMR19 Write Access Policy 1 (IMR19_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |
| 0 | 0x0<br>RW | **IMR19 Write Access Policy 0 (IMR19_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value. |

## 44.3.122 TPM Selector (B_CR_TPM_SELECTOR_0_0_0_MCHBAR)—Offset 6C24h

Specifies where the TPM lives in the platform. B-Unit uses this register to source decode for requests whose addresses fall within the fixed address range for the TPM.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6C24h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 1:0 | 0x0<br>RW | **TPM Selector (TPM_SELECTOR):**<br>• 0h: SoC doesn't support - fTPM enabled. Target of TPM accesses is the Intel ME.<br>• 1h: SPI TPM enabled. Target of TPM accesses is SPI.<br>• 2h: LPC TPM enabled. Target of TPM accesses is LPC.<br>• 3h: All TPMs disabled. Target of TPM accesses is the Error Handler. |

## 44.3.123 Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR)—Offset 6C80h

BIOS must write DEFVTDBAR and then immediately follow it up with a read to DEFVTDBAR to ensure that all copies of DEFVTDBAR in the system are updated.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 6C80h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:40 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 39 | 0x0<br>RO | **Reserved** |
| 38:12 | 0x0<br>RW | **Default IOMMU VTd Config Space Base (DEFVTDBAR):** If DEFVTDBAR is enabled, this field corresponds to bits 38:12 of the base address default IOMMU VTd configuration space. BIOS will program this register resulting in a base address for a 4KB block of contiguous memory address space.<br>This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space.<br>System Software uses this base address to program the default VTd IOMMU register set.<br>If DEFVTDBAR is enabled and incoming Request Address[38:12] matches DEFVTDBAR[38:12] the request targets the Default VTd BAR. |
| 11:2 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 1 | 0x0<br>RW | **Lock Register Content (LOCK):** Locks the contents of the register including itself. |
| 0 | 0x0<br>RW/L | **DEFVTDBAR Enable (DEFVTDBAREN):**<br>• 0: DEFVTDBAR is disabled and does not claim any memory<br>• 1: DEFVTDBAR memory mapped accesses are claimed and decoded appropriately. This bit will remain 0 if VTd capability is disabled. |

## 44.3.124 B-Unit Arbiter Control BARBCTRL0 (B_CR_BARBCTRL0)—Offset 6D4Ch

Specifies the weighting for Agents 03 used by the B-Unit's Badmit Arbiter. The value specified in the Agent Weight field is used by the Badmit arbiters weight counters to determine the number of requests from an agent that are allowed to be granted before updating the requester's age register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D4Ch

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 3 Weight (AGENT3_WEIGHT):** Arbiter weight for Agent 3. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 2 Weight (AGENT2_WEIGHT):** Arbiter weight for Agent 2. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 1 Weight (AGENT1_WEIGHT):** Arbiter weight for Agent 1. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 0 Weight (AGENT0_WEIGHT):** Arbiter weight for Agent 0. |

## 44.3.125 B-Unit Arbiter Control BARBCTRL1 (B_CR_BARBCTRL1)—Offset 6D50h

Specifies the weighting for Agents 4-7 used by the B-Unit's Badmit Arbiter. The value specified in the Agent Weight field is used by the Badmit arbiter's weight counters to determine the number of requests from an agent that are allowed to be granted before updating the requester's age register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D50h

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 7 Weight (AGENT7_WEIGHT):** Arbiter weight for Agent 7. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 6 Weight (AGENT6_WEIGHT):** Arbiter weight for Agent 6. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 5 Weight (AGENT5_WEIGHT):** Arbiter weight for Agent 5. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 4 Weight (AGENT4_WEIGHT):** Arbiter weight for Agent 4. |

## 44.3.126 B-Unit Scheduler Control (B_CR_BSCHWT0)—Offset 6D54h

Specifies the weighting for Agents 0-3 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter, to allow other agents' requests to be granted.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D54h

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 3 Weight (AGENT3_WEIGHT):** Arbiter weight for Agent 3. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 2 Weight (AGENT2_WEIGHT):** Arbiter weight for Agent 2. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 1 Weight (AGENT1_WEIGHT):** Arbiter weight for Agent 1. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 0 Weight (AGENT0_WEIGHT):** Arbiter weight for Agent 0. |

## 44.3.127 B-Unit Scheduler Control (B_CR_BSCHWT1)—Offset 6D58h

Specifies the weighting for Agents 4-7 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D58h

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 7 Weight (AGENT7_WEIGHT):** Arbiter weight for Agent 7. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 6 Weight (AGENT6_WEIGHT):** Arbiter weight for Agent 6. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 5 Weight (AGENT5_WEIGHT):** Arbiter weight for Agent 5. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 4 Weight (AGENT4_WEIGHT):** Arbiter weight for Agent 4. |

## 44.3.128 B-Unit Scheduler Control (B_CR_BSCHWT2)—Offset 6D5Ch

Specifies the weighting for Agents 8-11 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D5Ch

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 11 Weight (AGENT11_WEIGHT):** Arbiter weight for Agent 11. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 10 Weight (AGENT10_WEIGHT):** Arbiter weight for Agent 10. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 9 Weight (AGENT9_WEIGHT):** Arbiter weight for Agent 9. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 8 Weight (AGENT8_WEIGHT):** Arbiter weight for Agent 8. |

## 44.3.129 B-Unit Scheduler Control (B_CR_BSCHWT3)—Offset 6D60h

Specifies the weighting for Agents 12-15 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D60h

**Default:** 04040404h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 29:24 | 0x4<br>RW | **Agent 15 Weight (AGENT15_WEIGHT):** Arbiter weight for Agent 15. |
| 23:22 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 21:16 | 0x4<br>RW | **Agent 14 Weight (AGENT14_WEIGHT):** Arbiter weight for Agent 14. |
| 15:14 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 13:8 | 0x4<br>RW | **Agent 13 Weight (AGENT13_WEIGHT):** Arbiter weight for Agent 13. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_3):** Reserved. |
| 5:0 | 0x4<br>RW | **Agent 12 Weight (AGENT12_WEIGHT):** Arbiter weight for Agent 12. |

## 44.3.130 B-Unit Flush Control (B_CR_BWFLUSH)—Offset 6D64h

Controls the policy used to determine when dirty entries must be flushed to DRAM. When the number of dirty entries is lower than a high watermark dirty_hwm the B-Unit will opportunistically flush data to DRAM after the write flush timeout value. When the number of dirty entries exceeds the high water mark, the B-Unit will initiate a high priority flush and push dirty data to DRAM until the count is once again below the low water mark.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FF010000h

**BAR and Offset:** [MCHBAR] + 6D64h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0xff<br>RW | **Flush Threshold (FLUSH_THRESHOLD):** All write commands are blocked at Badmit if the number of write commands in the Flush Pool exceeeds this value. |
| 23:17 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 16 | 0x1<br>RO/V | **All Entries Flushed (ALL_ENTRIES_FLUSHED):** All dirty entries in the B-Unit, in both slices, have been flushed. |
| 15:8 | 0x0<br>RW | **Dirty Low Water Mark (DIRTY_LWM):** Low water mark for dirty entries retained by the B-Unit. B-Unit will immediately attempt to flush any dirty entry, hence setting the low water mark to 0. |
| 7:0 | 0x0<br>RW | **Dirty High Water Mark (DIRTY_HWM):** High water mark for dirty entries retained by the B-Unit. B-Unit will immediately attempt to flush any dirty entry, hence setting the low water mark to 0. |

## 44.3.131 B-Unit Flush Weights (B_CR_BFLWT)—Offset 6D68h

Controls B-Unit alternate scheduling of reads and writes to DRAM.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000404h

**BAR and Offset:** [MCHBAR] + 6D68h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Disable Flush Weights (DISABLE_FLUSH_WEIGHTS):** When set to 1, disables flush weights. Flushing of dirty entries will start when Dirty Limit is HWM, and continue until Dirty Limit is LWM. No reads will be scheduled in between. |
| 30:14 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 13:8 | 0x4<br>RW | **Write Weights (WRITE_WEIGHTS):** Number of write requests sent to a PMI channel before switching to scheduling read requests, when use of flush weights is not disabled. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 5:0 | 0x4<br>RW | **Read Weights (READ_WEIGHTS):** Number of read requests sent to a PMI channel before switching to scheduling write requests, when use of flush weights is not disabled. |

## 44.3.132 Weighted Scheduling Control of High Priority ISOC and Other Requests (B_CR_BISOCWT)—Offset 6D6Ch

Controls alternate scheduling of High Priority ISOC requests and other requests.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6D6Ch

**Default:** 80003F0Fh

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Enable ISOC Requests (ENABLE_ISOC_WEIGHTS):** When set to 1, enables switching from scheduling High Priority ISOC requests to scheduling Best Effort and Low Priority ISOC requests, based on ISOC weights and NONISOC weights. |
| 30:14 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 13:8 | 0x3f<br>RW | **ISOC Request Weights (ISOC_REQUEST_WEIGHTS):** Weight for high priority isochronous requests. |
| 7:6 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 5:0 | 0xf<br>RW | **Non ISOC Request Weights (NON_ISOC_REQUEST_WEIGHTS):** Weight for non-high priority isochronous and best effort requests. |

## 44.3.133 B-Unit Control (B_CR_BCTRL2)—Offset 6D70h

Contains basic control information used by the B-Unit.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000F0014h

**BAR and Offset:** [MCHBAR] + 6D70h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Enable Read Invalidate Timer (ENABLE_READ_INVALIDATE_TIMER):** When set to 1, enables the BRAM to clear the RD_DONE bit and flush dirty data when a timer expires.<br>Used when parity is not enabled, to force the B-Unit to not indefinitely cache previously read lines, and also to cause a flush of dirty data that has been written to the BRAM entry. |
| 30:25 | 0x0<br>RO | **Reserved (RESERVED_2):** Reserved. |
| 24 | 0x0<br>RW | **MOT Disable Stall Arbiter on Error (MOT_DISABLE_STALL_ARB_ON_ERR):** If the HH widget gets an error, allow transactions to get arbitrated. |
| 23:16 | 0xf<br>RW | **BRAM Read Invalidate Time (BRAM_READ_INVALIDATE_TIME):** Timer threshold to clear rd_done bits or flush a BRAM entry, if dirty.<br>The value specified is in multiples of 250ns. For each time interval specified, the read done status of one BRAM entry will be cleared based on a BTAG index which is then incremented to point to the next BRAM entry. |
| 15:8 | 0x0<br>RW | **Casual Timer (CASUAL_TIMER):** The number of clock cycles that the B-Unit waits before starting a casual dirty flush.<br>Casual Flush feature will not be enabled by PND2 architecture. Instead, a dirty write will be made eligible for scheduling immediately. |
| 7:5 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 4 | 0x1<br>RO | **Enable 64B Read (ENABLE_64B_READ):** When set, B-unit will send 64B PMI read requests for transactions requiring read access to DRAM.<br>Must always be set to one, otherwise functional errors may occur. Not applicable to the SoC, because the SoC is always 64 byte memory accesses. |
| 3 | 0x0<br>RW | **Demand Scrub Enable (DEMAND_SCRUB_ENABLE):** This mode causes B-Unit to automatically issue a flush to PMI, thus writing correct data back to memory for any read request that returns with a correctable data error. |
| 2 | 0x1<br>RW | **Enable Read Done for Write (ENABLE_READ_DONE_FOR_WRITE):** Enable Any writes (IWB or normal writes) to set the read_done bit in bstat, which enables return of data from the BRAM cache instead of Memory. |
| 1 | 0x0<br>RW | **Miss Valid Entries (MISS_VALID_ENTRIES):** This mode causes reads to clean valid B-Unit buffer entries to look like misses instead of hits.<br>When this bit is set -- even when all requested bytes are valid and present in the BRAM data buffer -- B-Unit will send read requests over the PMI interface to re-fetch the data from DRAM, instead of returning it from the BRAM. |
| 0 | 0x0<br>RW | **Dirty Stall (DIRTY_STALL):** This mode causes reads or writes from any requester interface to dirty valid B-Unit buffer entries to stall on the appropriate requester interface until the entry has been flushed from the B-Unit. |

## 44.3.134 Uncorrectable Error Status Register (B_CR_UNCERRSTS_0_0_0_MCHBAR)—Offset 6E04h

Errors that have been seen in bunit. The error is only logged if the respective bit in UNCERRMSK is 0.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E04h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RW/1C | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data<br><br>**Power Well:** prst |
| 8 | 0x0<br>RW/1C | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error<br><br>**Power Well:** prst |
| 7 | 0x0<br>RW/1C | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error<br><br>**Power Well:** prst |
| 6 | 0x0<br>RW/1C | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RW/1C | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory<br><br>**Power Well:** prst |
| 4 | 0x0<br>RW/1C | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space<br><br>**Power Well:** prst |
| 3 | 0x0<br>RW/1C | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO<br><br>**Power Well:** prst |
| 2 | 0x0<br>RW/1C | **MMIO HITM (MMIO_HITM):** MMIO Access HITM<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/1C | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW/1C | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected<br><br>**Power Well:** prst |

## 44.3.135 Uncorrectable Error Mask Register (B_CR_UNCERRMSK_0_0_0_MCHBAR)—Offset 6E08h

Masks whether a reported error is logged and signaled to the IEH. 1- do not log/signal. 0 - log/signal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E08h

**Default:** FFFFFFFFh

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Reserved** |
| 30 | 0x1<br>RO | **Reserved** |
| 29 | 0x1<br>RO | **Reserved** |
| 28 | 0x1<br>RO | **Reserved** |
| 27 | 0x1<br>RO | **Reserved** |
| 26 | 0x1<br>RO | **Reserved** |
| 25 | 0x1<br>RO | **Reserved** |
| 24 | 0x1<br>RO | **Reserved** |
| 23 | 0x1<br>RO | **Reserved** |
| 22 | 0x1<br>RO | **Reserved** |
| 21 | 0x1<br>RO | **Reserved** |
| 20 | 0x1<br>RO | **Reserved** |
| 19 | 0x1<br>RO | **Reserved** |
| 18 | 0x1<br>RO | **Reserved** |
| 17 | 0x1<br>RO | **Reserved** |
| 16 | 0x1<br>RO | **Reserved** |
| 15 | 0x1<br>RO | **Reserved** |
| 14 | 0x1<br>RO | **Reserved** |
| 13 | 0x1<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 12 | 0x1<br>RO | **Reserved** |
| 11 | 0x1<br>RO | **Reserved** |
| 10 | 0x1<br>RO | **Reserved** |
| 9 | 0x1<br>RW | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data |
| 8 | 0x1<br>RW | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error |
| 7 | 0x1<br>RW | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error |
| 6 | 0x1<br>RW | **Reserved** |
| 5 | 0x1<br>RW | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory |
| 4 | 0x1<br>RW | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space |
| 3 | 0x1<br>RW | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO |
| 2 | 0x1<br>RW | **MMIO HITM (MMIO_HITM):** MMIO Access HITM |
| 1 | 0x1<br>RW | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space |
| 0 | 0x1<br>RW | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected |

## 44.3.136 Uncorrectable Error Severity Register (B_CR_UNCERRSEV_0_0_0_MCHBAR)—Offset 6E0Ch

When set, escalate error as fatal to IEH, otherwise non-fatal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E0Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved** |
| 30 | 0x0<br>RO | **Reserved** |
| 29 | 0x0<br>RO | **Reserved** |
| 28 | 0x0<br>RO | **Reserved** |
| 27 | 0x0<br>RO | **Reserved** |
| 26 | 0x0<br>RO | **Reserved** |
| 25 | 0x0<br>RO | **Reserved** |
| 24 | 0x0<br>RO | **Reserved** |
| 23 | 0x0<br>RO | **Reserved** |
| 22 | 0x0<br>RO | **Reserved** |
| 21 | 0x0<br>RO | **Reserved** |
| 20 | 0x0<br>RO | **Reserved** |
| 19 | 0x0<br>RO | **Reserved** |
| 18 | 0x0<br>RO | **Reserved** |
| 17 | 0x0<br>RO | **Reserved** |
| 16 | 0x0<br>RO | **Reserved** |
| 15 | 0x0<br>RO | **Reserved** |
| 14 | 0x0<br>RO | **Reserved** |
| 13 | 0x0<br>RO | **Reserved** |
| 12 | 0x0<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x0<br>RO | **Reserved** |
| 10 | 0x0<br>RO | **Reserved** |
| 9 | 0x0<br>RW | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data |
| 8 | 0x0<br>RW | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error |
| 7 | 0x0<br>RW | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error |
| 6 | 0x0<br>RW | **Reserved** |
| 5 | 0x0<br>RW | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory |
| 4 | 0x0<br>RW | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space |
| 3 | 0x0<br>RW | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO |
| 2 | 0x0<br>RW | **MMIO HITM (MMIO_HITM):** MMIO Access HITM |
| 1 | 0x0<br>RW | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space |
| 0 | 0x0<br>RW | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected |

## 44.3.137 First Uncorrectable Error Register (B_CR_UNCFERRSTS_0_0_0_MCHBAR)—Offset 6E10h

First Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E10h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO/V | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO/V | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO/V | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO/V | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO/V | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory<br><br>**Power Well:** prst |
| 4 | 0x0<br>RO/V | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **MMIO HITM (MMIO_HITM):** MMIO Access HITM<br><br>**Power Well:** prst |
| 1 | 0x0<br>RO/V | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected<br><br>**Power Well:** prst |

## 44.3.138 Next Uncorrectable Error Register (B_CR_UNCNERRSTS_0_0_0_MCHBAR)—Offset 6E14h

Next Error Status

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E14h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO/V | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO/V | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO/V | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO/V | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO/V | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory<br><br>**Power Well:** prst |
| 4 | 0x0<br>RO/V | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO/V | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO/V | **MMIO HITM (MMIO_HITM):** MMIO Access HITM<br><br>**Power Well:** prst |
| 1 | 0x0<br>RO/V | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected<br><br>**Power Well:** prst |

## 44.3.139 Uncorrectable Error Select Register (B_CR_UNCERRCNTSEL_0_0_0_MCHBAR)—Offset 6E18h

Selects which errors cause the UNCERRCNT register to increment. 0-do not count 1-count.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E18h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved** |
| 30 | 0x0<br>RO | **Reserved** |
| 29 | 0x0<br>RO | **Reserved** |
| 28 | 0x0<br>RO | **Reserved** |
| 27 | 0x0<br>RO | **Reserved** |
| 26 | 0x0<br>RO | **Reserved** |
| 25 | 0x0<br>RO | **Reserved** |
| 24 | 0x0<br>RO | **Reserved** |
| 23 | 0x0<br>RO | **Reserved** |
| 22 | 0x0<br>RO | **Reserved** |
| 21 | 0x0<br>RO | **Reserved** |
| 20 | 0x0<br>RO | **Reserved** |
| 19 | 0x0<br>RO | **Reserved** |
| 18 | 0x0<br>RO | **Reserved** |
| 17 | 0x0<br>RO | **Reserved** |
| 16 | 0x0<br>RO | **Reserved** |
| 15 | 0x0<br>RO | **Reserved** |
| 14 | 0x0<br>RO | **Reserved** |
| 13 | 0x0<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 12 | 0x0<br>RO | **Reserved** |
| 11 | 0x0<br>RO | **Reserved** |
| 10 | 0x0<br>RO | **Reserved** |
| 9 | 0x0<br>RW | **Dram Read Data Uncorrectable Error (MC_RD_DATA_UNC):** CRC error in dram read data |
| 8 | 0x0<br>RW | **Bram Read Parity Error (BRAM_RD_PAR):** BRAM Read parity error |
| 7 | 0x0<br>RW | **Bram Write Parity Error (BRAM_WR_PAR):** BRAM Write parity error |
| 6 | 0x0<br>RW | **Reserved** |
| 5 | 0x0<br>RW | **IA HIT GSM (IA_HIT_GSM):** IA hit to graphics stolen memory |
| 4 | 0x0<br>RW | **MMIO CLFLUSH (MMIO_CLFLUSH):** ClFlush to MMIO address space |
| 3 | 0x0<br>RW | **MMIO WBMTOIE (MMIO_WBMTOIE):** WBMTo access to MMIO |
| 2 | 0x0<br>RW | **MMIO HITM (MMIO_HITM):** MMIO Access HITM |
| 1 | 0x0<br>RW | **PII request to MMIO (PII_2_MMIO):** NonIDI access to MMIO address space |
| 0 | 0x0<br>RW | **Unsupport IDI Opcode (UNS_IDI_OP):** Unsupport IDI Opcode detected |

## 44.3.140 Uncorrectable Error Count (B_CR_UNCERRCNT_0_0_0_MCHBAR)—Offset 6E1Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E1Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 15 | 0x0<br>RW/1C | **Error Count Overflow (OVERFLOW):** Reserved.<br><br>**Power Well:** prst |
| 14:0 | 0x0<br>RW/1C | **Uncorrectable Error Count (ERR_CNT):** Reserved.<br><br>**Power Well:** prst |

## 44.3.141 Correctable Error Status Register (B_CR_CORERRSTS_0_0_0_MCHBAR)—Offset 6E20h

Errors that have been seen in A-Unit. This will always log errors irrespective of the CORERRMSK bit setting

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E20h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 29 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 28 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 27 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 26 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 25 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 24 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 23 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 22 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 21 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 20 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 19 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 17 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 16 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 14 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 13 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 12 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 11 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 10 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 9 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 8 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 7 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 5 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 4 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 2 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **Reserved**<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW/1C | **MC_RD_DATA_COR:** MC_RD_DATA_COR<br><br>**Power Well:** prst |

## 44.3.142 Correctable Error Mask Register (B_CR_CORERRMSK_0_0_0_MCHBAR)—Offset 6E24h

Masks where a reported error is signaled to the IEH. 1- do not signal. 0 - signal.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 6E24h

**Default:** FFFFFFFFh

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Reserved** |
| 30 | 0x1<br>RO | **Reserved** |
| 29 | 0x1<br>RO | **Reserved** |
| 28 | 0x1<br>RO | **Reserved** |
| 27 | 0x1<br>RO | **Reserved** |
| 26 | 0x1<br>RO | **Reserved** |
| 25 | 0x1<br>RO | **Reserved** |
| 24 | 0x1<br>RO | **Reserved** |
| 23 | 0x1<br>RO | **Reserved** |
| 22 | 0x1<br>RO | **Reserved** |
| 21 | 0x1<br>RO | **Reserved** |
| 20 | 0x1<br>RO | **Reserved** |
| 19 | 0x1<br>RO | **Reserved** |
| 18 | 0x1<br>RO | **Reserved** |
| 17 | 0x1<br>RO | **Reserved** |
| 16 | 0x1<br>RO | **Reserved** |
| 15 | 0x1<br>RO | **Reserved** |
| 14 | 0x1<br>RO | **Reserved** |
| 13 | 0x1<br>RO | **Reserved** |
| 12 | 0x1<br>RO | **Reserved** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x1<br>RO | **Reserved** |
| 10 | 0x1<br>RO | **Reserved** |
| 9 | 0x1<br>RO | **Reserved** |
| 8 | 0x1<br>RO | **Reserved** |
| 7 | 0x1<br>RO | **Reserved** |
| 6 | 0x1<br>RO | **Reserved** |
| 5 | 0x1<br>RO | **Reserved** |
| 4 | 0x1<br>RO | **Reserved** |
| 3 | 0x1<br>RO | **Reserved** |
| 2 | 0x1<br>RO | **Reserved** |
| 1 | 0x1<br>RO | **Reserved** |
| 0 | 0x1<br>RW | **MC_RD_DATA_COR:** MC_RD_DATA_COR |

## 44.3.143 Data Parity Header Log Low (B_CR_DPHDRLOG_LO_0_0_0_MCHBAR)—Offset 6E28h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E28h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Location of Bad Data Byte (UNCDPLOG):** Reserved.<br><br>**Power Well:** prst |

## 44.3.144 Data Parity Header Log High (B_CR_DPHDRLOG_HI_0_0_0_MCHBAR)—Offset 6E2Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E2Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Location of Bad Data Byte (UNCDPLOG):** Reserved.<br><br>**Power Well:** prst |

## 44.3.145 Data Parity Address Log Low (B_CR_DPADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E30h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E30h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Data Parity Address (DPADDR):** Reserved.<br><br>**Power Well:** prst |

## 44.3.146 Data Parity Address Log High (B_CR_DPADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E34h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E34h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 7:0 | 0x0<br>RO/V | **Data Parity Address (DPADDR):** Reserved.<br><br>**Power Well:** prst |

## 44.3.147 D-Unit Error Address Log (B_CR_DERRADDRLOG_LO_0_0_0_MCHBAR)—Offset 6E38h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E38h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Address (ADDR):** D-Unit error address.<br><br>**Power Well:** prst |

## 44.3.148 D-Unit Error Address Log (B_CR_DERRADDRLOG_HI_0_0_0_MCHBAR)—Offset 6E3Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E3Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RO/V | **Data Parity Address (ADDR):** Reserved.<br><br>**Power Well:** prst |

## 44.3.149 Asymmetric Memory Region 0 (B_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR)—Offset 6E40h

Specification of asymmetric memory region 0 (in slice 0) for the configuration with 2 asymmetric memory regions.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E40h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Slice 0 Asymmetric Enable (SLICE0_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 0; setting it to 1 enables the region. |
| 30:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:20 | 0x0<br>RW | **SLICE0_ASYM_LIMIT:** Specifies bits [38:31] of the highest address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's highest address are equal to 1. |
| 19:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11:4 | 0x0<br>RW | **Slice 0 Asymmetric Limit (SLICE0_ASYM_BASE):** Specifies bits [38:31] of the base address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's base address are equal to 0. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 44.3.150 Asymmetric Memory Region 1 (B_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR)—Offset 6E44h

Specification of asymmetric memory region 1 (in slice 1) for the configuration with 2 asymmetric memory regions.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E44h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Slice 1 Asymmetric Enable (SLICE1_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 1; setting it to 1 enables the region. |
| 30:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:20 | 0x0<br>RW | **SLICE1_ASYM_LIMIT:** Specifies bits [38:31] of the highest address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's highest address are equal to 1. |
| 19:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11:4 | 0x0<br>RW | **Slice 1 Asymmetric Limit (SLICE1_ASYM_BASE):** Specifies bits [38:31] of the base address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's base address are equal to 0. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 44.3.151 B-Unit Machine Check Mode Low (B_CR_BMCMODE_LOW)—Offset 6E48h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000001h

**BAR and Offset:** [MCHBAR] + 6E48h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 0 | 0x1<br>RW | **Machine Check Signal Mode (MC_SIGNAL_MODE):** When set to 1, B-Unit will not allow any transaction with uncorrectable error or subsequent memory transaction to propagate through to Requester.<br>This will essentially hang CPU and CPU will end up with IERR shutdown.<br>Issue: When set to zero B-Unit will allow transaction with an uncorrectable error to propagate and signal MC event to CPU if enabled in IA32_MC5_CTL.<br>If enabled MC event will be taken by CPU cores at the end of instruction boundary after it detected by ROB<br><br>**Power Well:** prst |

## 44.3.152 B-Unit Machine Check Mode High (B_CR_BMCMODE_HIGH)—Offset 6E4Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 6E4Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

## 44.3.153 PWR_LIMIT_MISC_0_0_0_MCHBAR (P_CR_PWR_LIMIT_MISC_0_0_0_MCHBAR)—Offset 7014h

This register is Read Only. The register is used to report Skt power at Pn - mapped from FUSE_PN_POWER_OF_SKU and Minimal Time window - mapped from FUSE_PBM_MINIMAL_TAU_ALLOWED. Added new register for relaying IA PBM information to the use.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7014h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RO | **Reserved:** Reserved<br><br>**Power Well:** prst |
| 21:15 | 0x0<br>RO/V | **PBM_MINIMAL_TAU_ALLOWED:** Minimal Tau Time window allowed.<br><br>**Power Well:** prst |
| 14:0 | 0x0<br>RO/V | **PN_POWER_OF_SKU:** Package Power of SKU.<br><br>**Power Well:** prst |

## 44.3.154 CAPABILITY_ID_0_0_0_MCHBAR (P_CR_CAPABILITY_ID_0_0_0_MCHBAR)—Offset 7018h

This register is Read Only. The register is used to report to BIOS certain PM features for this Part.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7018h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved:** Reserved<br><br>**Power Well:** prst |
| 1:0 | 0x0<br>RO/V | **FUSE_THERMAL_PROFILE:**<br><br>**Power Well:** prst |

## 44.3.155 MC_BIOS_REQ_0_0_0_MCH (P_CR_MC_BIOS_REQ_0_0_0_MCHBAR)—Offset 7020h

This register is defined to support memSS reset flow, handshake between SOC FW (aka Pcode) and BIOS MRC

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7020h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **RUN_BUSY:** This bit indicates that the BIOS request is pending. BIOS sets this bit together with a command in the lower bits of this register. |
| 30:20 | 0x0<br>RW | **RSVD0:** Reserved - Future data bits for fastpath action or status indication. |
| 19 | 0x0<br>RW | **MEM_INIT_DONE:** BIOS programs this bit after DUNIT and DDRIO configuration is complete. |
| 18:16 | 0x0<br>RW | **Reserved** |
| 15 | 0x0<br>RW | **CH1_DIMMS:** BIOS writes this value indicates |
| 14 | 0x0<br>RW | **CH0_DIMMS:** BIOS writes this value indicates |
| 13 | 0x0<br>RW | **CH1_ENABLED:** BIOS writes this value indicates if memory channel 1 has device attached or not. 1'b1 = 1 or 2 DIMMs populated. |
| 12 | 0x0<br>RW | **CH0_ENABLED:** BIOS writes this value indicates if memory channel 1 has device attached or not. 1'b1 = 1 or 2 DIMMs populated. |
| 11:7 | 0x0<br>RW | **Reserved** |
| 6 | 0x0<br>RW | **DDR_TYPE_VOLTAGE:** BIOS set VDDQ Voltage: 0: DDR3 Memory Technology (VDDQ = 1.35V); 1: DDR4 Memory Technology (VDDQ = 1.25V); |
| 5:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.3.156 MCA_ERROR_SRC_0_0_0_MCHBAR (P_CR_MCA_ERROR_SRC_0_0_0_MCHBAR)—Offset 702Ch

This register logs error source information i.e IERR or MCERR information for Pcode. The error fields are cleared by HW or BIOS.This register is also shadowed in the I/O space.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 702Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/V | **CATERR:** Asserted by HW on IERR or MCERR assertion.<br><br>**Power Well:** prst |
| 30 | 0x0<br>RW/V | **IERR:** Asserted by HW on IERR assertion.<br><br>**Power Well:** prst |
| 29 | 0x0<br>RW/V | **MCERR:** Asserted by HW on MCERR assertion.<br><br>**Power Well:** prst |
| 28:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RW | **RESERVED:**<br><br>**Power Well:** prst |

## 44.3.157 DDR_THERM_THRT_CTRL_0_0_0_MCHBAR (P_CR_DDR_THERM_THRT_CTRL_0_0_0_MCHBAR)—Offset 7030h

This register is used to configure thermal throttling policies for both near and far memory.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00010000h

**BAR and Offset:** [MCHBAR] + 7030h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **RESERVED** |
| 16 | 0x1<br>RW/V | **RESERVED** |
| 15:10 | 0x0<br>RO | **RESERVED** |
| 9 | 0x0<br>RW/V | **MEM_THRT_ENABLE:** When set, memory traffic is throttled if near Wide-IO or far LPDDR memory MR4 value = FM/NM_THERM_THRT_THRESHOLD, respectively. Thermal throttling is achieved by applying memory bandwidth clips in the memory subsystem. |
| 8 | 0x0<br>RW/V | **MEM_THRT_CFG:** Configure memory throttling behavior. Policies are defined as follows:<br>• 0 = Thermal throttling policy uses instantaneous MR4 status for FM/ NM_THERM_THRT_THRESHOLD<br>• 1 = Thermal throttling policy uses time filtered MR4 status for FM/NM_THERM_THRT_THRESHOLD. Filtering time constant is configured in the DDR_THERM_INTERRUPT register. |
| 7 | 0x0<br>RW/V | **NM_THERM_THRT_ENABLE:** When set, throttling is activated if near Wide I/O memory MR4 value is greater than or equal to NM_THERM_THRT_THRESHOLD |
| 6:4 | 0x0<br>RW/V | **NM_THERM_THRT_THRESHOLD:** Configurable threshold of near Wide I/O memory MR4 value greater than or equal to which thermal throttling is activated.<br>• Near Memory MR4 (Threshold = Enable thermal throttling)<br>• Near Memory MR4 (Threshold = Disable thermal throttling) |
| 3 | 0x0<br>RW/V | **FM_THERM_THRT_ENABLE:** When enabled throttling is activated if far LPDDR memory MR4 value is greater than or equal to FM_THERM_THRT_THRESHOLD.<br>For the standard LPDDR DRAM that is only capable up to Tcasemax of 85C this bit should be set to enable far LPDDR throttling to keep DRAM within its Tcasemax spec. |
| 2:0 | 0x0<br>RW/V | **FM_THERM_THRT_THRESHOLD:** Configurable threshold of far LPDDR memory MR4 value greater than or equal to which thermal throttling is activated.<br>For the standard LPDDR DRAM that is only capable up to Tcasemax of 85C this field should be configured to avoid DRAM to exceed its Tcasemax spec.<br>• Far Memory MR4 (Threshold = Enable thermal throttling)<br>• Far Memory MR4 (Threshold = Disable thermal throttling) |

## 44.3.158 PACKAGE_THERM_MARGIN (P_CR_PACKAGE_THERM_MARGIN_0_0_0_MCHBAR)—Offset 703Ch

Temperature margin in PECI temperature counts from the thermal profile specification. Platform fan control SW is expected to read therm_margin value to control fan or blower speed.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 703Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **RESERVED** |
| 15:0 | 0x0<br>RW | **THERM_MARGIN:** Temperature margin in PECI temperature counts from the thermal profile specification.<br>THERM_MARGIN is in 2's complement format (8.8 format where MSB equals 1 Sign bit + 7 bits of integer temperature value and the LSB equals 8 precision bits of temperature value).<br>A value of zero indicates the hottest CPU die temperature is on the thermal profile line.<br>A negative value indicates gap to the thermal profile that platform SW should increase cooling capacity.<br>A sustained negative value should be avoided as it may impact part reliability. |

## 44.3.159 DDR_RAPL_LIMIT (P_CR_DDR_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 7040h

RAPL power limit for DDR domain. This register is written by platform software and read by Pcode once per 1ms. There are actually two instances of this register: one with MSR access and one with MMIO access. There is a separate PECI/PCS command which is also analogous.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 7040h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/L | **LOCKED:** When set this entire register becomes read-only. This bit will typically be set by BIOS during boot. |
| 62:32 | 0x0<br>RO | **Reserved:** Reserved |
| 31:24 | 0x0<br>RO | **Reserved:** Reserved |
| 23:22 | 0x0<br>RW/L | **LIMIT1_TIME_WINDOW_X:** The time window for RAPL is expressed in a format called TAU. TAU is specified in a mantissa+exponent format.<br>This format is common across all RAPL interfaces for TAU. The format is specified with a 7 bit number that looks like: XXYYYYY (2 bits of mantissa followed by 5 exponent)<br>The time window corresponding to this config is calculated as: (1+ XXb / 4) * (2^YYYYYb) The value that results from this math species the time window in units that are defined in PACKAGE_POWER_SKU_UNIT.TIME_UNIT MSR. |
| 21:17 | 0x0<br>RW/L | **LIMIT1_TIME_WINDOW_Y:** The Tau Exponent. See TIME_WINDOW_X for more information. |
| 16 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 15 | 0x0<br>RW/L | **LIMIT1_ENABLE:** Power Limit[0] enable bit for DDR domain. |
| 14:0 | 0x0<br>RW/L | **LIMIT1_POWER:** Power Limit[0] for DDR domain. The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. |

## 44.3.160 DDR_ENERGY_STATUS (P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7048h

Reports total energy consumed in DRAM. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. This energy status is what is used by DDR RAPL or OLTM control algorithms if the product supports those features.

To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7048h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **JOULES:** Total Joules of energy consumed by all DIMMs. Units are proportional to Joules and are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT.<br><br>**Power Well:** prst |

## 44.3.161 DDR_RAPL_PERF_STATUS (P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 704Ch

Memory RAPL performance throttling counter. This counter accumulates time that any channel in the memory controller is bandwidth throttled due to memory RAPL constraints. This counter counts total time (in PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT units) that any channel is throttled. If two channels are throttled, this counter increments at a 2x rate, so that for 1ms in wall clock time the counter counts 2ms. This counter does not include throttling as a result of thermal management or MEMHOT. This register is updated at approximately 1ms intervals. This counter is normalized to 'seconds' and is not subject to variation of actual DRAM clock speeds. This register is read-only for software via MMIO MSR and PECI/PCS. This register starts counting at zero from reset and continues counting forever and wraparounds may occur, so software should ensure the sample rate is sufficient to avoid aliasing. This is an unsigned value.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 704Ch

**Default:** 000058F0h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x58f0<br>RW | **DURATION:** Bandwidth throttle duration counter due to Memory RAPL. Sum across all channels in PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT units. This data can serve as a proxy for the potential performance impacts of RAPL on memory accesses.<br><br>**Power Well:** prst |

## 44.3.162 PACKAGE_RAPL_PERF_STATUS (P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR)—Offset 7050h

Counts time that any core in the IA domain is performance throttled below OS request and below the base frequency (P1) because of power limits (PL1 or PL2). Counts in time units defined by PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT. If software uses the TURBO_ACTIVATION_RATIO or PECI ACPI P-NOTIFY, the turbo activation ratios described by those features will may elevate the effective OS request (as calculated by this counter) to max turbo. This register starts counting at zero from reset and continues counting forever and wraparounds may occur, so software should ensure the sample rate is sufficient to avoid aliasing. This is an unsigned value.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7050h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **COUNTS:** Time that any core in the IA domain is performance throttled below OS request and below the base frequency (P1) because of power limits (PL1 or PL2).<br>Counts in time units defined by PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT<br><br>**Power Well:** prst |

## 44.3.163 PRIMARY_PLANE_TURBO_POWER_POLICY (P_CR_PRIMARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7054h

The PRIMARY_PLANE_TURBO_POWER_POLICY and SECONDARY_PLANE_TURBO_POWER_POLICY are used together as hints to balance the power budget between the primary (IA core) and secondary (Graphics) power planes. This biasing is effectively a performance biasing, and it helps Punit firmware assess where software needs performance the most.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7054h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 4:0 | 0x0<br>RW | **PRIPTP:** Performance priority Level for the IA Core (primary) power plane.<br>A higher number implies a higher priority. |

## 44.3.164 SECONDARY_PLANE_TURBO_POWER_POLICY (P_CR_SECONDARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR)—Offset 7058h

The PRIMARY_PLANE_TURBO_POWER_POLICY and SECONDARY_PLANE_TURBO_POWER_POLICY are used together as hints to balance the power budget between the primary (IA core) and secondary (Graphics) power planes. This biasing is effectively a performance biasing, and it helps Punit firmware assess where software needs performance the most.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7058h

**Default:** 00000010h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 4:0 | 0x10<br>RW | **SECPTP:** Performance priority Level for the Graphics (secondary) power plane.<br>A higher number implies a higher priority. |

## 44.3.165 PRIMARY_PLANE_ENERGY_STATUS (P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 705Ch

Reports total energy consumed across all IA cores. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms.

To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 705Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **DATA:** Contains an accumulated value of the energy consumed in the primary power plane.<br>To find the energy consumed in a given time window, software should subtract the two energy readings.<br>Software will have to take care of counter wrapping around when it overflows.<br>Units are proportional to Joules exact precision is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT |

## 44.3.166 SECONDARY_PLANE_ENERGY_STATUS (P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 7060h

Reports total energy consumed across all IA cores. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms.

To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7060h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **DATA:** Contains an accumulated value of the energy consumed in the secondary power plane.<br>To find the energy consumed in a given time window, software should subtract the two energy readings.<br>Software will have to take care of counter wrapping around when it overflows.<br>Units are proportional to Joules exact precision is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT |

## 44.3.167 PACKAGE_POWER_SKU_UNIT (P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR)—Offset 7068h

Defines units for calculating SKU power, current, energy, resistance and timing parameters.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 330A0E08h

**BAR and Offset:** [MCHBAR] + 7068h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x3<br>RW | **RESISTANCE_UNIT:** Used to define the units of resistance for control registers that describe parameters in ohms such as VR_CURRENT_CONFIG.<br>The actual unit value is calculated by 1mohm / 2^RESISTANCE_UNIT.<br>The default value of 3 corresponds to 0.125mohm. |
| 27:24 | 0x3<br>RW | **CURRENT_UNIT:** Used to define the units of amps in control registers such as VR_CURRENT_CONFIG.<br>The actual unit value is calculated by 1A / 2^CURRENT_UNIT.<br>The default value of 3 corresponds to 0.125A. |
| 23:20 | 0x0<br>RO | **RESERVED_2:** Reserved |
| 19:16 | 0xa<br>RW | **TIME_UNIT:** Used for to define the time units in registers such as PL1, PL2, PL3 and PL4.<br>The actual unit value is calculated by 1s / 2^TIME_UNIT.<br>The default value of 10 corresponds to 0.977ms. |
| 15:13 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 12:8 | 0xe<br>RW | **ENERGY_UNIT:** Used to define the units of energy reporting registers such as PACKAGE_ENERGY_STATUS.<br>The actual unit value is calculated by 1 J / 2^ENERGY_UNIT.<br>The default value of 14 corresponds to ~61uJ per bit. |
| 7:4 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 3:0 | 0x8<br>RW | **PWR_UNIT:** Used to define the units of power control registers such as PL1, PL2, PL3 and PL4.<br>The actual unit value is calculated by 1 W / 2^PWR_UNIT.<br>The default value of 8 corresponds to 3.9mW per bit. |

## 44.3.168 PACKAGE_ENERGY_STATUS (P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR)—Offset 706Ch

Reports total energy consumed across the entire SOC / Package. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. This energy status is what is used by RAPL PL1, PL2 and PL3 control algorithms.

To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 706Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **DATA:** Contains accumulated energy consumed by the entire CPU.<br>This counter will wrap around and keep counting when the counter overflows.<br>Units are proportional to Joules exact precision is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT |

## 44.3.169 GT_PERF_STATUS (P_CR_GT_PERF_STATUS_0_0_0_MCHBAR)—Offset 7070h

Contains the voltage and ratio status for GT. This register is mapped to GT_PERF_STATUS_0_0_0_MCHBAR.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7070h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:26 | 0x0<br>RO | **RSVD0:** Reserved |
| 25:17 | 0x0<br>RO/V | **RP_STATE_RATIO_SLICE:** Ratio of the current RP-state, in 16.6Mhz 1xclks.<br>When the graphics engine is in RC6, this field is zeroed out.<br>For the SoC the Unslice and Slice run at the same ratio. |
| 16:8 | 0x0<br>RO/V | **RP_STATE_RATIO_UNSLICE:** Ratio of the current RP-state, in 16.6Mhz 1xclks.<br>When the graphics engine is in RC6, this field is zeroed out.<br>For the SoC the Unslice and Slice run at the same ratio. |
| 7:0 | 0x0<br>RO/V | **RP_STATE_VOLTAGE:** RP-State Voltage GT Target Voltage in U1.7 Volts |

## 44.3.170 TEMPERATURE_TARGET (P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR)—Offset 7074h

This register contains information about the fan speed control target temperature as well as details on the reference temperature for IA core DTS relative temperature reading.

**MSR_Name**: TEMPERATURE_TARGET **MSR_Addr**: 0x1A2

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7074h

**Default:** 005A0000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 30:24 | 0x0<br>RW | **TJ_MAX_TCC_OFFSET:** This field allows platform software to configure the temperature at which thermal monitor engages to be lower than the manufacturing configured maximum constraint.<br>This field is programmed in 1'C units. E.g., if the default silicon configured maximum temperature is 100'C and this field is configured to 10, then the silicon will engage thermal throttling algorithms at 90'C |
| 23:16 | 0x5a<br>RO/V | **REF_TEMP:** Tjmax a.k.a. Thermal Monitor activation temperature or Prochot Temperature.<br>This is the maximum junction temperature at which thermal throttling aka thermal monitor is activated.<br>This temperature is the maximum temperature at which the silicon is capable of operating at.<br>All IA core digital thermal sensor readings are reported as a relative negative offset from this reference temperature, such that a read-on of zero implies the cores are running at this temperature. |
| 15:8 | 0x0<br>RO/V | **FAN_TEMP_TARGET_OFFSET:** Fan Temperature Target Offset a.k.a. TControl indicates the relative offset from the the Thermal Monitor Trip Temperature at which fans should be engaged. |
| 7:0 | 0x0<br>RO | **RESERVED_0:** Reserved |

## 44.3.171 BIOS_RESET_CPL (P_CR_BIOS_RESET_CPL_0_0_0_MCHBAR)—Offset 7078h

This register is used as a means for BIOS to communicate staging to the Punit / Pcode. The exact definition and utility of each bit may differ across products. The general philosophy is that when BIOS is done with stage0, it writes the RST_CPL0 bit and then waits for the PCODE_INIT_DONE0 bit to be set before proceeding to the next step.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7078h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **RESERVED0:** reserved |
| 15 | 0x0<br>RO/V | **PCODE_INIT_DONE7:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 14 | 0x0<br>RO/V | **PCODE_INIT_DONE6:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 13 | 0x0<br>RO/V | **PCODE_INIT_DONE5:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 12 | 0x0<br>RO/V | **PCODE_INIT_DONE4:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 11 | 0x0<br>RO/V | **PCODE_INIT_DONE3:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 10 | 0x0<br>RO/V | **PCODE_INIT_DONE2:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 9 | 0x0<br>RO/V | **PCODE_INIT_DONE1:** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 8 | 0x0<br>RO/V | **PCODE_INIT_DONE:** Pcode has completed its actions in response to Stage0 BIOS Reset complete.<br>Between BIOS Stage0 complete and Pcode Stage0 complete, Pcode will apply all power savings configurations to PCS and will set up C_STATE_LATENCY control MSR settings for IRTL management. |
| 7 | 0x0<br>RW | **RST_CPL7:** reset complete |
| 6 | 0x0<br>RW | **RST_CPL6:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 5 | 0x0<br>RW | **RST_CPL5:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 4 | 0x0<br>RW | **RST_CPL4:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 3 | 0x0<br>RW | **RST_CPL3:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **RST_CPL2:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 1 | 0x0<br>RW | **RST_CPL1:** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage.<br>Stage validity is product-specific and this stage may not be applicable to this product. |
| 0 | 0x0<br>RW | **RST_CPL:** Set by BIOS to indicate that all power management configurations as part of reset are complete.<br>This must include Punit patch load done as well as all relevant Punit power management register and mailbox configurations done.<br>Once this bit is set, Punit will allow normal power management to start.<br>Before setting this bit, P-states and C-states support is disabled. BIOS should wait before receiving the Pcode Stage0 reset complete before proceeding with any further steps. |

## 44.3.172 BIOS_MAILBOX_DATA (P_CR_BIOS_MAILBOX_DATA_0_0_0_MCHBAR)—Offset 7080h

Data register for the BIOStoPCODE mailbox. This mailbox is implemented as a means for accessing statistics and implementing BIOS-Pcode handshakes. This register is used in conjunction with BIOS_MAILBOX_INTERFACE.

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7080h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **DATA:** This field contains the data associated with specific commands. |

## 44.3.173 BIOS_MAILBOX_INTERFACE (P_CR_BIOS_MAILBOX_INTERFACE_0_0_0_MCHBAR)—Offset 7084h

Control and Status register for the BIOS to Pcode mailbox. This mailbox is implemented as a means for accessing statistics and implementing BIOS-Pcode handshakes. This register is used in conjunction with BIOS_MAILBOX_DATA.

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7084h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S/V | **RUN_BUSY:** SW may write to the two mailbox registers only when RUN_BUSY is clear(0).<br>Setting RUN_BUSY to 1 will pend a Fast Path event to Pcode.<br>After setting this bit SW will poll this bit until it is cleared.<br>Pcode will clear RUN_BUSY after updating the mailbox registers with the result and error code. |
| 30:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:8 | 0x0<br>RW/V | **ADDRESS:** This field is used to specify an additional parameter to extend the command when needed. |
| 7:0 | 0x0<br>RW/V | **COMMAND:** This field contains the SW request command or the Pcode response code depending on the setting of RUN_BUSY. |

## 44.3.174 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7088h

PUNIT_MMIO: Core Frequency Capabilities This register describes the frequency capabilities of the IA cores. Units are 100MHz multiplied by the ratio. Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the processing system frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 7088h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **LAST_RESOLVED_FREQ:** Last resolved ratio for the IA cores.<br>Units are 100MHz multiplied by the ratio.<br>This value is updated dynamically whenever the IA core frequency changes. |
| 23:16 | 0x0<br>RO/V | **MAX_SUPPORTED_FREQ:** Maximum ratio for the IA cores.<br>Units are 100MHz multiplied by the ratio. |
| 15:8 | 0x0<br>RO/V | **EFFICIENT_FREQ:** Firmware-calculated efficient ratio for the IA cores.<br>Units are 100MHz multiplied by the ratio. |
| 7:0 | 0x0<br>RO/V | **MIN_SUPPORTED_FREQ:** Minimum supported ratio for the IA cores.<br>Units are 100MHz multiplied by the ratio. |

## 44.3.175 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 708Ch

PUNIT_MMIO: Graphics Engine Frequency Capabilities This register describes the frequency capabilities of the integrated graphics engine. Units are 50MHz multiplied by the ratio. Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the integrated graphics engine frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 708Ch

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **LAST_RESOLVED_FREQ:** Last resolved ratio for the integrated graphics engine.<br>Units are 50MHz multiplied by the ratio.<br>This value is updated dynamically whenever the graphics engine frequency changes. |
| 23:16 | 0x0<br>RO/V | **MAX_SUPPORTED_FREQ:** Maximum supported ratio for the integrated graphics engine.<br>Units are 50MHz multiplied by the ratio. |
| 15:8 | 0x0<br>RO/V | **EFFICIENT_FREQ:** Firmware-calculated efficient ratio for the integrated graphics engine.<br>Units are 50MHz multiplied by the ratio. |
| 7:0 | 0x0<br>RO/V | **MIN_SUPPORTED_FREQ:** Minimum supported ratio for the integrated graphics engine.<br>Units are 50MHz multiplied by the ratio. |

## 44.3.176 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7090h

PUNIT_MMIO: System Agent Frequency Capabilities This register describes the frequency capabilities of the System Agent. Units are 16.666MHz multiplied by the ratio. Last resolved ratio is updated upon changes to the System Agent frequency.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 7090h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0 RO/V | **LAST_RESOLVED_RATIO:** Last resolved System Agent ratio, in units of 16.666MHz. |
| 23:16 | 0x0 RO/V | **RESERVED_2:** Reserved |
| 15:8 | 0x0 RO/V | **RESERVED_1:** Reserved |
| 7:0 | 0x0 RO/V | **RESERVED_0:** Reserved |

## 44.3.177 FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7094h

This register reports out the DDR far memory frequency. The actual capabilities of the SOC with respect to DDR frequency is described in the MEMSS_FREQUENCY_CAPABILITIES register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 7094h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0 RO/V | **LAST_RESOLVED_RATIO:** This field reports out the DDR far memory frequency in integer multiple. This register reflects what BIOS has programmed as the default DDR frequency |
| 23:16 | 0x0 RO/V | **RESERVED_2:** Reserved |
| 15:8 | 0x0 RO/V | **RESERVED_1:** Reserved |
| 7:0 | 0x0 RO/V | **RESERVED_0:** Reserved |

## 44.3.178 NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_NEAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7098h

This register reports out the Wide I/O near memory frequency. A value of zero indicates there is no Wide I/O DRAM present. The actual capabilities of the SOC with respect to Wide I/O frequency is described in the MEMSS_FREQUENCY_CAPABILITIES register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 7098h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **LAST_RESOLVED_RATIO:** This field reports out the Wide I/O near memory frequency in integer multiple of 133.33MHz.<br>This register reflects what BIOS has programmed as the default Wide I/O near memory frequency |
| 23:16 | 0x0<br>RO/V | **RESERVED_2:** Reserved |
| 15:8 | 0x0<br>RO/V | **RESERVED_1:** Reserved |
| 7:0 | 0x0<br>RO/V | **RESERVED_0:** Reserved |

## 44.3.179 PACKAGE_POWER_SKU (P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR)—Offset 70A0h

This register describes the power SKU of the part and limits on time window and power limit configuration allowed in the RAPL PL1 and PL2 configuration registers.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0012024000600118h

**BAR and Offset:** [MCHBAR] + 70A0h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:55 | 0x0<br>RO | **RESERVED_3:** Reserved |
| 54:48 | 0x12<br>RW | **PKG_MAX_WIN:** The maximal time window allowed to be programmed for RAPL PL1 and PL2 controls for the SKU.<br>Higher values will be clamped to this value.<br>The bits of this field describe parameters for a mathematical equation for time window configuration.<br>This field is split into two sub-fields:<br>• x = bits[6:5]<br>• y = bits[4:0]<br>Time window equation:<br>**time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**<br><br>**Power Well:** prst |
| 47 | 0x0<br>RO | **RESERVED_2:** Reserved |
| 46:32 | 0x240<br>RW | **PKG_MAX_PWR:** The maximal package power setting allowed for the SKU.<br>Higher values will be clamped to this value.<br>The maximum setting is typical not guaranteed.<br>The default value for this field is determined by fuses.<br>The units for this value are defined in PACKAGE_POWER_SKU_MSR[PWR_UNIT].<br><br>**Power Well:** prst |
| 31 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 30:16 | 0x60<br>RW | **PKG_MIN_PWR:** The minimal package power setting allowed for the SKU.<br>Lower values may not be achievable by run-time RAPL PL1 and PL2 control algorithms.<br><br>**Power Well:** prst |
| 15 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 14:0 | 0x118<br>RW | **PKG_TDP:** The TDP package power setting allowed for the SKU.<br>The units for this value are defined in PACKAGE_POWER_SKU_MSR[PWR_UNIT]. |

## 44.3.180 PACKAGE_RAPL_LIMIT (P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR)—Offset 70A8h

Package RAPL Power Limit allows a software agent to define power limitation for the package domain. Power limitation is defined in terms of average power usage (Watts) over a time window specified. Two power limits and associated time windows can be specified. These power limits are commonly referred to as PL1 (long time window) and PL2 (short time window). Each power limit provides independent clamping control that would permit the processor cores to go below OS-requested state to meet the power limits. A lock mechanism allow the software agent to enforce power limit settings. Once the lock bit is set, the power limit settings are static and un-modifiable until next RESET.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 70A8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/L | **PKG_PWR_LIM_LOCK:** When set all settings in this register are locked and are treated as Read Only.<br>This lock control is persistent until the next reset.<br>This bit will typically set by BIOS during boot time or resume from Sx. |
| 62:56 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 55:49 | 0x0<br>RW/L | **PKG_PWR_LIM_2_TIME:** Time window for Power Limit 1 (PL2).<br>This describes the control window of the power limit.<br>This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s.<br>The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN.<br>There is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective.<br>The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields:<br>• x = bits[6:5]<br>• y = bits[4:0]<br>Time window equation:<br>**time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)** |
| 48 | 0x0<br>RW/VL | **PKG_CLMP_LIM_2:** Clamp mode control for PL2.<br>• 0 = PL2 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC.<br>• 1 = PL2 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level.<br>In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled. |
| 47 | 0x0<br>RW/VL | **PKG_PWR_LIM_2_EN:** Enable for Power Limit 2 (PL2).<br>Setting this bit activates the power limit and time window defined for PL2. |
| 46:32 | 0x0<br>RW/L | **PKG_PWR_LIM_2:** Sets the average power usage limit of the package domain corresponding to the PL2 time window.<br>The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT.<br>This power limit must be configured by software before it will engage.<br>The PL2 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures. |
| 31:24 | 0x0<br>RO | **RESERVED_0:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 23:17 | 0x0<br>RW/L | **PKG_PWR_LIM_1_TIME:** Time window for Power Limit 1 (PL1).<br>This describes the control window of the power limit.<br>This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s.<br>The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN.<br>There is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective.<br>The bits of this field describe parameters for a mathematical equation for time window configuration.<br>This field is split into two sub-fields:<br>• x = bits[6:5]<br>• y = bits[4:0]<br>Time window equation:<br>**time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)** |
| 16 | 0x0<br>RW/VL | **PKG_CLMP_LIM_1:** Clamp mode control for PL1.<br>• 0 = PL1 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC.<br>• 1 = PL1 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level.<br>In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled. |
| 15 | 0x0<br>RW/VL | **PKG_PWR_LIM_1_EN:** Enable for Power Limit 1 (PL1).<br>Setting this bit activates the power limit and time window defined for PL1. |
| 14:0 | 0x0<br>RW/L | **PKG_PWR_LIM_1:** Sets the average power usage limit of the package domain corresponding to the PL1 time window.<br>The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT.<br>This power limit must be configured by software before it will engage.<br>The PL1 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures. |

## 44.3.181 IA_PERF_LIMIT_REASONS (P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR)—Offset 70B0h

This register reports reasons for performance limitations on the IA cores. Status bits are an instantaneous indication of an active constraint. Log bits indicate that a constraint was enforced since the log bit was last cleared.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 70B0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/0C/V | **QOS_LOG:** Logged indication that frequency was clamped below the software-defined quality-of-service floor.<br>This bit is set by firmware, and is clearable by software. |
| 30 | 0x0<br>RW/0C/V | **MAX_EFFICIENCY_FREQ_LOG:** Logged indication that frequency was clamped below the firmware-calculated maximum efficiency frequency.<br>This bit is set by firmware, and is clearable by software. |
| 29 | 0x0<br>RW/0C/V | **MCT_LOG:** Logged indication that frequency was clamped due to ratio change transition attenuation.<br>This bit is set by firmware, and is clearable by software. |
| 28 | 0x0<br>RW/0C/V | **EDP_LOG:** Logged indication that frequency was clamped due to the package-level Electrical Design Point constraint.<br>This bit is set by firmware, and is clearable by software. |
| 27 | 0x0<br>RW/0C/V | **MULTI_CORE_TURBO_LOG:** Logged indication that frequency was clamped due to effective multi-core turbo constraints.<br>This bit is set by firmware, and is clearable by software. |
| 26 | 0x0<br>RW/0C/V | **VR_THERMALERT_LOG:** Logged indication that frequency was clamped due to a voltage regulator thermal excursion.<br>This bit is set by firmware, and is clearable by software. |
| 25 | 0x0<br>RW/0C/V | **IA_UTILIZATION_LOG:** Logged indication that frequency was clamped due to the autonomous utilization-based P-state control algorithm.<br>This bit is set by firmware, and is clearable by software. |
| 24 | 0x0<br>RW/0C/V | **DEV3_LOG:** Logged indication that frequency was clamped due to a Device 3 driver override.<br>This bit is set by firmware, and is clearable by software. |
| 23 | 0x0<br>RW/0C/V | **DEV2_LOG:** Logged indication that frequency was clamped due to a Device 2 driver override.<br>This bit is set by firmware, and is clearable by software. |
| 22 | 0x0<br>RW/0C/V | **SPARE6_LOG:** Spare log bit.<br>This bit is set by firmware, and is clearable by software. |
| 21 | 0x0<br>RW/0C/V | **SPARE5_LOG:** Spare log bit.<br>This bit is set by firmware, and is clearable by software. |
| 20 | 0x0<br>RW/0C/V | **SPARE4_LOG:** Spare log bit.<br>This bit is set by firmware, and is clearable by software. |
| 19 | 0x0<br>RW/0C/V | **PL2_LOG:** Logged indication that frequency was clamped due to a package-level PL2 excursion.<br>This bit is set by firmware, and is clearable by software. |
| 18 | 0x0<br>RW/0C/V | **PL1_LOG:** Logged indication that frequency was clamped due to a package-level PL1 excursion.<br>This bit is set by firmware, and is clearable by software. |
| 17 | 0x0<br>RW/0C/V | **THERMAL_LOG:** Logged indication that frequency was clamped due to a thermal excursion.<br>This bit is set by firmware, and is clearable by software. |
| 16 | 0x0<br>RW/0C/V | **PROCHOT_LOG:** Logged indication that frequency was clamped due to PROCHOT assertion.<br>This bit is set by firmware, and is clearable by software. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO/V | **QOS_STATUS:** Frequency is limited below the operating system or driver Quality-of-Service floor. |
| 14 | 0x0<br>RO/V | **MAX_EFFICIENCY_FREQ_STATUS:** Frequency is limited below the maximum efficiency frequency. |
| 13 | 0x0<br>RO/V | **MCT_STATUS:** Frequency is limited due to ratio change transition attenuation (MCT, prevents frequent ratio changes due to core C-state entry/exit). |
| 12 | 0x0<br>RO/V | **EDP_STATUS:** Frequency is limited due to a package-level EDP constraint. |
| 11 | 0x0<br>RO/V | **MULTI_CORE_TURBO_STATUS:** Frequency is limited due to effective multi-core turbo constraints. |
| 10 | 0x0<br>RO/V | **VR_THERMALERT_STATUS:** Frequency is limited due to a VR thermal excursion. |
| 9 | 0x0<br>RO/V | **IA_UTILIZATION_STATUS:** Frequency is limited due to autonomous utilization-based P-state control. |
| 8 | 0x0<br>RO/V | **DEV3_STATUS:** Frequency is limited due to Dev3 driver override. |
| 7 | 0x0<br>RO/V | **DEV2_STATUS:** Frequency is limited due to Dev2 driver override. |
| 6 | 0x0<br>RO/V | **SPARE6_STATUS:** Spare status bit. |
| 5 | 0x0<br>RO/V | **SPARE5_STATUS:** Spare status bit. |
| 4 | 0x0<br>RO/V | **SPARE4_STATUS:** Spare status bit. |
| 3 | 0x0<br>RO/V | **PL2_STATUS:** Frequency is limited due to a package-level PL2 excursion. |
| 2 | 0x0<br>RO/V | **PL1_STATUS:** Frequency is limited due to a package-level PL1 excursion. |
| 1 | 0x0<br>RO/V | **THERMAL_STATUS:** Frequency is limited due to thermal excursion. |
| 0 | 0x0<br>RO/V | **PROCHOT_STATUS:** Frequency is limited due to external PROCHOT assertion. |

## 44.3.182 MEMHOT_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_STATUS_0_0_0_MCHBAR)—Offset 70D8h

Memhot indications and logging

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 70D8h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **RESERVED:** Reserved |
| 1 | 0x0<br>RW/0C/V | **MEMHOT_LOG:** Set to 1 on MEMHOT_N pin assertion; remains set until cleared via CR write of 1 value or chip reset. |
| 0 | 0x0<br>RO/V | **MEMHOT_STATUS:**<br>• 0 = MEMHOT_N pin not asserted<br>• 1 = MEMHOT_N pin is asserted |

## 44.3.183 MEMHOT_THERM_CONFIG_0_0_0_MCHBAR (P_CR_MEMHOT_THERM_CONFIG_0_0_0_MCHBAR)—Offset 70DCh

Configuration data for memhot.

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 70DCh

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **LOCK:** When 1, writes to other MEMHOT_THERM_CONFIG bits are not allowed |
| 30:2 | 0x0<br>RO | **RESERVED:** Reserved |
| 1:0 | 0x0<br>RW/L | **MEMHOT_THROT_LVL:**<br>• 00 = MEMHOT disabled (no response to pin assertion)<br>• 01 = THRT_MID<br>• 10 = THRT_HI<br>• 11 = THRT_CRIT |

## 44.3.184 MEM_THERM_CTRL_0_0_0_MCHBAR (P_CR_MEM_THERM_CTRL_0_0_0_MCHBAR)—Offset 7200h

Configuration data for OLTT, CLTT, and MEMTRIP

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7200h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:8 | 0x0<br>RW | **TEMP_MEMTRIP:** Temperature at which mem trip will be asserted |
| 7:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5 | 0x0<br>RO/V | **SMBUSACKD_FLAG:** Set by CLTT SMBus controller when DIMM temperatures are ready to be read. This bit is also used as the valid bit for the DIMM temperatures.<br>Pcode will clear this bit when initiating a new DIMM temperature read request. |
| 4:3 | 0x0<br>RW | **OLTT_THROT_LVL:** Throttle level when OLTT is enabled, reuses CLTT throttle levels.<br>• 00 = no throttle<br>• 01 = THRT_MID<br>• 10 = THRT_HI<br>• 11 = THRT_CRIT |
| 2 | 0x0<br>RW | **CLTT_MODE:**<br>• 0 = CLTT pass-thru<br>• 1 = CLTT over SMBus (ignore if MTM_MODE = 0) |
| 1 | 0x0<br>RW | **MEM_THERM_MGMT_MODE:** Memory thermal mgmt. mode:<br>• 0 = OLTT,<br>• 1 = CLTT |
| 0 | 0x0<br>RW | **EN_MEMTRIP:** Mem_thermtrip control.<br>• 0: THERMTRIP is CPU thermal sensor only<br>• 1: THERMTRIP is CPU thermal sensor or mem_thermtrip |

## 44.3.185 MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_CONFIG_0_0_0_MCHBAR)—Offset 7204h

CLTT temperature thresholds

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7204h

**Default:** 00645D54h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 26:24 | 0x0<br>RW | **TEMP_THROT_HYST:** Negative-going threshold hysteresis value.<br>This value is subtracted from the TEMP_* threshold values to determine the point where the asserted status for that threshold will clear as temperature decreases.<br>TEMP_THROT_HYST must be at least 1C less than the minimum delta between adjacent TEMP_* thresholds. |
| 23:16 | 0x64<br>RW | **TEMP_HI:** High threshold temperature, 8.8.0ufp.<br>Program so TEMP_HI ) TEMP_MID. |
| 15:8 | 0x5d<br>RW | **TEMP_MID:** Middle threshold temperature, 8.8.0ufp.<br>Program so TEMP_MID ( TEMP_HI. |
| 7:0 | 0x54<br>RW | **TEMP_LO:** Lower threshold temperature, 8.8.0ufp.<br>Program so TEMP_LO ( TEMP_MID. |

## 44.3.186 MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR (P_CR_MEM_THERM_THROT_CONFIG_0_0_0_MCHBAR)—Offset 7208h

CLTT bandwidth throttle levels

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7208h

**Default:** 00635A00h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 23:16 | 0x63<br>RW | **THRT_CRIT:** Throttle level in %BW (= 100, 8.8.0ufp. Default is 99% throttled. |
| 15:8 | 0x5a<br>RW | **THRT_HI:** Throttle level in %BW ( THRT_CRIT, 8.8.0ufp (default is 90% throttled) |
| 7:0 | 0x0<br>RW | **THRT_MID:** Throttle level in %BW ( THRT_HI , 8.8.0ufp (default is 0% throttled) |

## 44.3.187 MEM_THERM_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_STATUS_0_0_0_MCHBAR)—Offset 720Ch

CLTT per-DIMM status/log info

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 720Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RESERVED_3:** Reserved |
| 27:26 | 0x0<br>RW/0C/V | **MEM_THERM_THROT_LOG_1_1:** Sticky bits set to highest throttle level achieved (same encoding as MEM_THERM_STATUS bits) when CLTT memory throttling is initiated by channel1, DIMM1; remains set until cleared via CR write of 1 value or chip reset. |
| 25:24 | 0x0<br>RO/V | **MEM_THERM_STATUS_1_1:** Status of channel1, DIMM1: 00 = memory temperature ( TEMP_LO 01 = TEMP_LO (= mem temp ( TEMP_MID 10 = TEMP_MID (= mem temp ( TEMP_HI 11 = mem temp )= TEMP_HI |
| 23:20 | 0x0<br>RO | **RESERVED_2:** Reserved |
| 19:18 | 0x0<br>RW/0C/V | **MEM_THERM_THROT_LOG_1_0:** Sticky bits set to highest throttle level achieved (same encoding as MEM_THERM_STATUS bits) when CLTT memory throttling is initiated by channel1, DIMM0; remains set until cleared via CR write of 1 value or chip reset. |
| 17:16 | 0x0<br>RO/V | **MEM_THERM_STATUS_1_0:** Status of channel1, DIMM0: 00 = memory temperature ( TEMP_LO 01 = TEMP_LO (= mem temp ( TEMP_MID 10 = TEMP_MID (= mem temp ( TEMP_HI 11 = mem temp )= TEMP_HI |
| 15:12 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 11:10 | 0x0<br>RW/0C/V | **MEM_THERM_THROT_LOG_0_1:** Sticky bits set to highest throttle level achieved (same encoding as MEM_THERM_STATUS bits) when CLTT memory throttling is initiated by channel0, DIMM1; remains set until cleared via CR write of 1 value or chip reset. |
| 9:8 | 0x0<br>RO/V | **MEM_THERM_STATUS_0_1:** Status of channel0, DIMM1: 00 = memory temperature ( TEMP_LO 01 = TEMP_LO (= mem temp ( TEMP_MID 10 = TEMP_MID (= mem temp ( TEMP_HI 11 = mem temp )= TEMP_HI |
| 7:4 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 3:2 | 0x0<br>RW/0C/V | **MEM_THERM_THROT_LOG_0_0:** Sticky bits set to highest throttle level achieved (same encoding as MEM_THERM_STATUS bits) when CLTT memory throttling is initiated by channel0, DIMM0; remains set until cleared via CR write of 1 value or chip reset. |
| 1:0 | 0x0<br>RO/V | **MEM_THERM_STATUS_0_0:** Status of channel0, DIMM0: 00 = memory temperature ( TEMP_LO 01 = TEMP_LO (= mem temp ( TEMP_MID 10 = TEMP_MID (= mem temp ( TEMP_HI 11 = mem temp )= TEMP_HI |

## 44.3.188 MEM_THERM_INTERRUPT_0_0_0_MCHBAR (P_CR_MEM_THERM_INTERRUPT_0_0_0_MCHBAR)—Offset 7210h

CLTT and customer-configurable interrupt configuration

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7210h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 23 | 0x0<br>RW | **RESERVED_2:** Reserved |
| 22:16 | 0x0<br>RW | **RESERVED_3:** Reserved |
| 15 | 0x0<br>RW | **MEMHOT_SMI_ENABLE:** Enable thermal interrupt generation whenever maximum DIMM_adj_TEMP_x_y value crosses MEM_INT_THRESHOLD1_VALUE |
| 14:8 | 0x0<br>RW | **MEM_INT_THRESHOLD1_VALUE:** Configurable memory temperature threshold1 value for thermal interrupt generation, unsigned integer U7.7.0 |
| 7:3 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 2 | 0x0<br>RW | **TEMP_HI_SMI_ENABLE:** Enable interrupt generation whenever maximum DIMM_adj_TEMP_x_y value crosses TEMP_HI |
| 1 | 0x0<br>RW | **TEMP_MID_SMI_ENABLE:** Enable interrupt generation whenever maximum DIMM_adj_TEMP_x_y value crosses TEMP_MID |
| 0 | 0x0<br>RW | **TEMP_LOW_SMI_ENABLE:** Enable interrupt generation whenever maximum DIMM_adj_TEMP_x_y value crosses TEMP_LO |

## 44.3.189 MEM_THERM_INT_STATUS_0_0_0_MCHBAR (P_CR_MEM_THERM_INT_STATUS_0_0_0_MCHBAR)—Offset 7214h

CLTT and customer-configurable interrupt status and logs

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7214h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/0C/V | **CLTT_SMBUS_HANG_LOG:** Set to 1 the first time CLTT_SMBUS_HANG_STATUS is set and remains sticky until cleared via software or reset. |
| 30 | 0x0<br>RO/V | **CLTT_SMBUS_HANG_STATUS:** Indication that the CLTT SMBUS is hung |
| 29 | 0x0<br>RW/0C/V | **CLTT_SMBUS_RESET_LOG:** Set to 1 the first time CLTT_SMBUS_RESET_STATUS is set and remains sticky until cleared via software or reset. |
| 28 | 0x0<br>RO/V | **CLTT_SMBUS_RESET_STATUS:** Indication that the CLTT SMBUS is reset |
| 27:14 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 13 | 0x0<br>RW/0C/V | **Reserved** |
| 12 | 0x0<br>RO/V | **Reserved** |
| 11 | 0x0<br>RW/0C/V | **SOC_MEM_THROTTLE_LOG:** Set to 1 the first time SOC_MEM_THROTTLE_STATUS is set and remains sticky until cleared via software or reset |
| 10 | 0x0<br>RO/V | **SOC_MEM_THROTTLE_STATUS:** Temperature is at or above the point to cause SoC memory throttling |
| 9 | 0x0<br>RW/0C/V | **RESERVED_LOG:** Reserved |
| 8 | 0x0<br>RO/V | **RESERVED_STATUS:** Reserved |
| 7 | 0x0<br>RW/0C/V | **MEMHOT_SMI_LOG:** Set to 1 the first time MEM_INT_THRESHOLD1_STATUS is set and remains sticky until cleared via software or reset. |
| 6 | 0x0<br>RO/V | **MEMHOT_SMI_STATUS:** Maximum DIMM_adj_TEMP_x_y value is at or above MEM_INT_THRESHOLD1_VALUE |
| 5 | 0x0<br>RW/0C/V | **TEMP_HI_SMI_LOG:** Set to 1 the first time TEMP_HI_SMI_STATUS is set and remains sticky until cleared via software or reset. |
| 4 | 0x0<br>RO/V | **TEMP_HI_SMI_STATUS:**<br>• 1: MTM is at the TEMP_HI throttle state.<br>• 0: MTM is below the TEMP_HI throttle state. |
| 3 | 0x0<br>RW/0C/V | **TEMP_MID_SMI_LOG:** Set to 1 the first time TEMP_MID_SMI_STATUS is set and remains sticky until cleared via software or reset. |
| 2 | 0x0<br>RO/V | **TEMP_MID_SMI_STATUS:**<br>• 1: MTM is at the TEMP_MID throttle state.<br>• 0: MTM is below the TEMP_MID throttle state. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/0C/V | **TEMP_LO_SMI_LOG:** Set to 1 the first time TEMP_LO_SMI_STATUS is set and remains sticky until cleared via software or reset. |
| 0 | 0x0<br>RO/V | **TEMP_LO_SMI_STATUS:**<br>• 1: MTM is at or above the TEMP_LO throttle state.<br>• 0: MTM is in the unthrottled state. |

## 44.3.190 MEM_THERM_TEMP_OFST_0_0_0_MCHBAR (P_CR_MEM_THERM_TEMP_OFST_0_0_0_MCHBAR)—Offset 7218h

CLTT per-DIMM temperature offsets

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7218h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RESERVED_3:** Reserved |
| 28:24 | 0x0<br>RW | **MEM_TEMP_OFST_DDR_1_1:** 4-bit integer temperature offset value for channel1, DIMM1 TSOD. |
| 23:21 | 0x0<br>RO | **RESERVED_2:** Reserved |
| 20:16 | 0x0<br>RW | **MEM_TEMP_OFST_DDR_1_0:** 4-bit integer temperature offset value for channel1, DIMM0 TSOD. |
| 15:13 | 0x0<br>RO | **RESERVED_1:** Reserved |
| 12:8 | 0x0<br>RW | **MEM_TEMP_OFST_DDR_0_1:** 4-bit integer temperature offset value for channel0, DIMM1 TSOD. |
| 7:5 | 0x0<br>RO | **RESERVED_0:** Reserved |
| 4:0 | 0x0<br>RW | **MEM_TEMP_OFST_DDR_0_0:** 4-bit integer temperature offset value for channel0, DIMM0 TSOD. |

## 44.3.191 TELEM_NEAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_NEAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E0h

This counter measures the total time spent with near memory active, as measured by any rank being in the active or active idle state. The inverse of this counter indicates the total time spent with all near memory in the self-refresh state. This counter counts at the crystal clock frequency divided by 16.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 70E0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V | **DATA:** This counter measures the total time spent with near memory active, as measured by any rank being in the active or active idle state.<br>The inverse of this counter indicates the total time spent with all near memory in the self-refresh state.<br>This counter counts at the crystal clock frequency divided by 16. |

## 44.3.192 TELEM_FAR_MEMORY_ACTIVE_ACCUMULATOR (P_CR_TELEM_FAR_MEMORY_ACTIVE_0_0_0_MCHBAR)—Offset 70E8h

This counter measures the total time spent with far memory active, as measured by any rank being in the active or active idle state. The inverse of this counter indicates the total time spent with all far memory in the self-refresh state. This counter counts at the crystal clock frequency divided by 16.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 70E8h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V | **DATA:** This counter measures the total time spent with far memory active, as measured by any rank being in the active or active idle state.<br>The inverse of this counter indicates the total time spent with all far memory in the self-refresh state.<br>This counter counts at the crystal clock frequency divided by 16. |

## 44.3.193 TURBO_ACTIVATION_RATIO (P_CR_TURBO_ACTIVATION_RATIO_0_0_0_MCHBAR)—Offset 70F0h

PUNIT_MSR: Used by software to configure the P-state at which turbo activates. This feature is functionally identical to the ACPI P-state notify supported over PECI (PCS 33). Turbo activation starts at clock ratios above the turbo activation ratio threshold configured in this register.

**MSR_Name**: TURBO_ACTIVATION_RATIO **MSR_Addr**: 0x64C

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 70F0h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **TURBO_ACTIVATION_RATIO_LOCK:** Write to 1b to lock the setting in this register until next reset |
| 30:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RW/L | **MAX_NON_TURBO_RATIO:** The P-unit will treat any P-state request above this ratio as a request for maximum turbo. Zero is special encoding which disables the feature |

## 44.3.194 C2C3TT_CFG (P_CR_C2C3TT_CFG_0_0_0_MCHBAR)—Offset 7034h

DEPRECATED. This register contains the initial snoop timer (pop-down) value. BIOS can update this value during run-time. Pcode will sample this register at slow loop. If the value has changed since the previous sample and in addition there is no valid Hysteresis parameter (HYS) from a previous PM_DMD or PM_RSP message, then Pcode will configure IMPH_CR_SNP_RELOAD[LIM] with this value. This register is not used by the SoC.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7034h

**Default:** 00000032h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:0 | 0x32<br>RW | **PPDN_INIT:** Value |

## 44.3.195 DTS_CONFIG1_CFG (P_CR_DTS_CONFIG1_CFG_0_0_0_MCHBAR)—Offset 7110h

DEPRECATED. DTS 2.0 is not supported. Location of correction factor F and scale factor C for DTS2.0

bscherka: IVT HSD i1112231 DTS2.0 added this csr isteiner: Moved DTS registers for re-convergence

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7110h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0 RW/L | **RSVD1:** Reserved |
| 28:16 | 0x0 RW/L | **C:** Scale factor C May be unlocked for select OEMs |
| 15:13 | 0x0 RW/L | **RSVD:** reserved |
| 12:0 | 0x0 RW/L | **F:** Correction Factor F May be unlocked for select OEMs |

## 44.3.196 DTS_CONFIG2_CFG (P_CR_DTS_CONFIG2_CFG_0_0_0_MCHBAR)—Offset 7114h

DEPRECATED. DTS 2.0 is not supported. Location of time constant coefficients for implementation.

bscherka: IVT HSD i1112231 added this csr isteiner: Moved DTS registers for re-convergence

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7114h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0 RW/L | **RSVD1:** Reserved be4538196 - attached to correct fuse |
| 28:16 | 0x0 RW/L | **AS:** Time constant coefficient AS May be unlocked for select OEMs |
| 15:13 | 0x0 RW/L | **RSVD:** reserved |
| 12:0 | 0x0 RW/L | **AF:** Time constant coefficient AF May be unlocked for select OEMs |

## 44.3.197 DTS_CONFIG3_CFG (P_CR_DTS_CONFIG3_CFG_0_0_0_MCHBAR)—Offset 7118h

Location of Tcontrol_offset for DTS2.0

bscherka: IVT added HSD i1112231 isteiner: Moved DTS registers for re-convergence

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7118h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28 | 0x0<br>RW/L | **OFFSET_PROGRAMMED:** Bit that the offset was programmed. |
| 27:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7:0 | 0x0<br>RW/L | **TCONTROL_OFFSET:** Offset to modify Tcontrol for DTS2.0 |

## 44.3.198 SOUTH_IO_PORT_STATUS3 (P_CR_SOUTH_IO_PORT_STATUS3_0_0_0_MCHBAR)—Offset 711Ch

Power meter data for south IPs

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 711Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:26 | 0x0<br>RW | **RSVD:** reserved |
| 25:24 | 0x0<br>RW | **GBE_PORTS:** Snapshot of the number of GBE lanes for power meter. |
| 23:16 | 0x0<br>RW | **PCIE_LANES:** Snapshot of the number of PCIE lanes for power meter. |
| 15:8 | 0x0<br>RW | **SATA_LANES:** Snapshot of the number of SATA lanes for power meter. |
| 7:4 | 0x0<br>RW | **USB3_LANES:** Snapshot of the number of USB3 lanes for power meter. |
| 3:0 | 0x0<br>RW | **USB2_LANES:** Snapshot of the number of USB2 lanes for power meter. |

## 44.3.199 DRAM_BIOS_INFO (P_CR_DRAM_BIOS_INFO_0_0_0_MCHBAR)—Offset 7128h

This register provides DRAM interface information for initial boot and SPD warm reset. It is populated by Pcode and read by BIOS.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7128h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **Reserved:** Reserved |
| 23:16 | 0x0<br>RW | **CURRENT_VID_VALUE:** Current voltage of VDDQ SVID Rail. |
| 15:12 | 0x0<br>RW | **RSVD1:** Reserved |
| 11 | 0x0<br>RW | **SPD_RESET_STATUS:** This field indicates to BIOS that a SPD Warm Reset has already occurred. |
| 10:8 | 0x0<br>RW | **DDR_CURRENT_FREQ:** This field indicates the current DDR IO clock frequency. |
| 7:4 | 0x0<br>RW | **RSVD2:** Reserved |
| 3 | 0x0<br>RW | **Reserved:** Reserved |
| 2:0 | 0x0<br>RW | **DDR_MAX_FREQ_LIMIT:** This field indicates to BIOS the Maximum supported DDR frequency for this product.<br>Pcode populates this information from fuses.<br>For the DDR_MAX_Freg_Limit, the value always shows 4 (100b).<br>• 000b = 1600 MHz<br>• 001b = 1833 MHz<br>• 010b = 2133 MHz<br>• 011b = 2400 MHz<br>• 100b = 2666 MHz |

## 44.3.200 ADR_COMMAND (P_CR_ADR_COMMAND_0_0_0_MCHBAR)—Offset 712Ch

This register provides PMC Firmware the ability to trigger an ADR (Asynchronous DRAM Self-Refresh). Fast path event for Pcode to take action and flush context information to DRAMs on imminent power loss.

THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 712Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:1 | 0x0<br>RW/V | **Reserved:** Reserved |
| 0 | 0x0<br>RW/V | **L2_FLUSH:** Flush the L2 cache on all modules as part of the ADR flow. |

## 44.3.201 DRAM_POWER_INFO (P_CR_DRAM_POWER_INFO_0_0_0_MCHBAR)—Offset 7130h

PUNIT_MSR: MSR for DRAM power information **MSR_Name**: DRAM_POWER_INFO
**MSR_Addr**: 0x61C

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 7130h

**Default:** 0028000000000118h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/L | **LOCK:** The Spec power allowed for DRAM. The TDP setting is typical (not guaranteed). |
| 62:55 | 0x0<br>RO | **RSVD0:** Reserved |
| 54:48 | 0x28<br>RW/L | **DRAM_MAX_WIN:** The maximal time window allowed for the DRAM.<br>Higher values will be clamped to this value. x = PKG_MAX_WIN[54:53], y = PKG_MAX_WIN[52:48].<br>The timing interval window is Floating Point number given by 1.x * power(2,y). |
| 47 | 0x0<br>RO | **RSVD1:** Reserved |
| 46:32 | 0x0<br>RW/L | **DRAM_MAX_PWR:** The maximal power setting allowed for DRAM.<br>Higher values will be clamped to this value. The maximum setting is typical (not guaranteed). |
| 31 | 0x0<br>RO | **RSVD2:** Reserved |
| 30:16 | 0x0<br>RW/L | **DRAM_MIN_PWR:** The minimal power setting allowed for DRAM.<br>Lower values will be clamped to this value. The minimum setting is typical (not guaranteed). |
| 15 | 0x0<br>RO | **RSVD3:** Reserved |
| 14:0 | 0x118<br>RW/L | **DRAM_TDP:** The Spec power allowed for DRAM.<br>The TDP setting is typical (not guaranteed). |

## 44.3.202 PCODE_WRITE_SPARE (P_CR_PCODE_WRITE_SPARE_0_0_0_MCHBAR)—Offset 7138h

Spare register for future use

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7138h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **SPARE_BITS:** spare bits for stepping ID use |

## 44.3.203 PACKAGE_TEMPERATURES_0_0_0_MCHBAR (P_CR_PACKAGE_TEMPERATURES_0_0_0_MCHBAR)—Offset 70F4h

Read-only register used for monitoring thermal status from all domains in the package.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 70F4h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **SA_Temperature:** System agent domain max temperature in degrees C.<br>Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0).<br>Raw, unfiltered |
| 23:16 | 0x0<br>RO/V | **Reserved:** Reserved |
| 15:8 | 0x0<br>RO/V | **Reserved:** Reserved |
| 7:0 | 0x0<br>RO/V | **IA_Temperature:** Virtual max temperature of all IA cores in degrees C.<br>Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0).<br>Raw, unfiltered |

## 44.3.204 MEMSS_FREQUENCY_CAPABILITIES (P_CR_MEMSS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR)—Offset 7108h

Describes the minimum and maximum frequency capabilities of LPDDR and WideIO on this particular SOC. If the maximum supported frequency reports a zero for LPDDR or WideIO, it indicates that the respective DRAM technology is not supported on this product.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7108h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RW | **NUM_NM_CH:** Reserved. |
| 29:24 | 0x0<br>RW | **WIO_FREQ:** Reserved. |
| 23:18 | 0x0<br>RW | **LP4_FREQ_HIGH:** Reserved. |
| 17:12 | 0x0<br>RW | **LP4_FREQ_LOW:** Reserved. |
| 11:6 | 0x0<br>RW | **LP3_FREQ_HIGH:** Reserved. |
| 5:0 | 0x0<br>RW | **LP3_FREQ_LOW:** Reserved. |

## 44.3.205 PP1_C0_CORE_CLOCK_0_0_0_MCHBAR (P_CR_PP1_C0_CORE_CLOCK_0_0_0_MCHBAR)—Offset 7160h

GT RC0 residency counter. Holds the accumulated number of CS clks that GT has been in RC0.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7160h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **DATA:** Accumulated cycles GT has been in RC0. |

## 44.3.206 Core Exists Vector (P_CR_CORE_EXISTS_VECTOR_0_0_0_MCHBAR)—Offset 7164h

Indication of the physical presence of IA cores in this silicon. IA core modules are defined as containing pairs of cores and an associated L2 cache. Module existence can therefore be inferred by OR'ing pairs of COREx_EXISTS in this register. This register does not reflect the impact of any software-based core disabling. It always reflects the capabilities of the silicon.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7164h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW | **CORE15_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 14 | 0x0<br>RW | **CORE14_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 13 | 0x0<br>RW | **CORE13_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 12 | 0x0<br>RW | **CORE12_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 11 | 0x0<br>RW | **CORE11_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 10 | 0x0<br>RW | **CORE10_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 9 | 0x0<br>RW | **CORE9_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 8 | 0x0<br>RW | **CORE8_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 7 | 0x0<br>RW | **CORE7_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 6 | 0x0<br>RW | **CORE6_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 5 | 0x0<br>RW | **CORE5_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 4 | 0x0<br>RW | **CORE4_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 3 | 0x0<br>RW | **CORE3_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 2 | 0x0<br>RW | **CORE2_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 1 | 0x0<br>RW | **CORE1_EXISTS:** A 1 indicates this core is enabled in the SoC. |
| 0 | 0x0<br>RW | **CORE0_EXISTS:** A 1 indicates this core is enabled in the SoC. |

## 44.3.207 Software Core Disable Mask (P_CR_CORE_DISABLE_MASK_0_0_0_MCHBAR)—Offset 7168h

Software may disable cores using this interface. The bit definition of this register exactly matches that defined in the CORE_EXISTS_VECTOR register. The flow is as follows:

- Cold boot
- BIOS reads CORE_EXISTS_VECTOR to establish which modules and cores are present
- BIOS writes 1b to the corresponding core that it wishes to **disable**.
- BIOS may disable entire modules by writing 1b to a pair of cores.
- The results of the configuration are maintained in the sustain power well.
- BIOS initiates a cold reset flow at the platform. In a cold reset, the sustain power well maintains power most other rails are power cycled.
- On cold reset exit, launches only the cores requested by BIOS. To software, it will appear as if these cores do not exist.

Software may discover the resolved core exists vector: resolved_core_exists_vector = (!CORE_DISABLE_MASK) and CORE_EXISTS_VECTOR

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7168h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW | **CORE15_DISABLE_MASK:** core15 disable mask<br><br>**Power Well:** prst |
| 14 | 0x0<br>RW | **CORE14_DISABLE_MASK:** core14 disable mask<br><br>**Power Well:** prst |
| 13 | 0x0<br>RW | **CORE13_DISABLE_MASK:** core13 disable mask<br><br>**Power Well:** prst |
| 12 | 0x0<br>RW | **CORE12_DISABLE_MASK:** core12 disable mask<br><br>**Power Well:** prst |
| 11 | 0x0<br>RW | **CORE11_DISABLE_MASK:** core11 disable mask<br><br>**Power Well:** prst |
| 10 | 0x0<br>RW | **CORE10_DISABLE_MASK:** core10 disable mask<br><br>**Power Well:** prst |
| 9 | 0x0<br>RW | **CORE9_DISABLE_MASK:** core9 disable mask<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8 | 0x0<br>RW | **CORE8_DISABLE_MASK:** core8 disable mask<br><br>**Power Well:** prst |
| 7 | 0x0<br>RW | **CORE7_DISABLE_MASK:** core7 disable mask<br><br>**Power Well:** prst |
| 6 | 0x0<br>RW | **CORE6_DISABLE_MASK:** core6 disable mask<br><br>**Power Well:** prst |
| 5 | 0x0<br>RW | **CORE5_DISABLE_MASK:** core5 disable mask<br><br>**Power Well:** prst |
| 4 | 0x0<br>RW | **CORE4_DISABLE_MASK:** core4 disable mask<br><br>**Power Well:** prst |
| 3 | 0x0<br>RW | **CORE3_DISABLE_MASK:** core3 disable mask<br><br>**Power Well:** prst |
| 2 | 0x0<br>RW | **CORE2_DISABLE_MASK:** core2 disable mask<br><br>**Power Well:** prst |
| 1 | 0x0<br>RW | **CORE1_DISABLE_MASK:** core1 disable mask<br><br>**Power Well:** prst |
| 0 | 0x0<br>RW | **CORE0_DISABLE_MASK:** core0 disable mask<br><br>**Power Well:** prst |

## 44.3.208 PL3_CONTROL_MCHBAR (P_CR_PL3_CONTROL_0_0_0_MCHBAR)—Offset 71F0h

Control Power Limit 3 (PL3) and Power Limit 4 (PL4) using this register. This limit control is physically different from the same control in the IA core MSR space.

- PL3 is designed to clamp peak sustained power to levels supported by the battery or input power supply and as such manage lifetime degradation of that power delivery element. With PL3, peak power excursions above the limit are allowed so long as they do not exceed the configured duty cycle constraint in this register.
- PL4 is designed to clamp peak instantaneous power to levels below the max supported by the battery or input power supply. These clamps are implemented a priori and the SOC is guaranteed to constrain itself below the PL4 limit always.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 71F0h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/L | **LOCK:** Write a 1b to lock this register until next reset.<br>Once locked, no further updates may be written to any bits in the register. |
| 62:48 | 0x0<br>RO | **RSVD0:** Reserved |
| 47 | 0x0<br>RW/L | **PL4_ENABLE:**<br>• 0 = disabled<br>• 1 = enabled |
| 46:32 | 0x0<br>RW/L | **PMAX:** Power Limit 'PL4' or Pmax power limit in the units as described PACAKGE_POWER_SKU_UNIT MSR.<br>The SoC guarantees it will never exceed this power limit even for very short time windows. |
| 31 | 0x0<br>RO | **RSVD1:** Reserved |
| 30:24 | 0x0<br>RW/L | **DUTY_CYCLE:** Power limit excursion duty cycle control for PL3, describing what percentage of time it is allowed for the SOC to exceed the programmed PL3 power limit.<br>0% implies excursions are not supported ever and 100% implies excursions are always allowed (effectively disabling the feature).<br>Units are in percentage(%). E.g., to allow for 20% excursion time and 80% PL3 power limit clamp time, program a value of 14h.<br>Values greater than 100 (64h) are clipped to 100%. |
| 23:17 | 0x0<br>RW/L | **TIME_WINDOW:** Duration over which duty cycle control will be maintained.<br>The bits of this field describe parameters for a mathematical equation for time window configuration.<br>This time window is strictly adhered to, if the window described is 40ms, then silicon guarantees no excursions to the programmed duty cycle within a rolling 40ms window.<br>This field is split into two sub-fields:<br>• x = bits[6:5]<br>• y = bits[4:0]<br>Time window equation:<br>**time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)** |
| 16 | 0x0<br>RO | **RSVD2:** Reserved |
| 15 | 0x0<br>RW/L | **PL3_ENABLE:**<br>• 0 = disabled<br>• 1 = enabled |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14:0 | 0x0<br>RW/L | **POWER_LIMIT:** Power Limit 3 (PL3) or PAppMax power level.<br>Any SoC power measurement observed above this level is considered as an excursion against the PL3 power limit and duty cycle / time window budget.<br>Units of this power limit are defined by PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. |

## 44.3.209 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL1_0_0_0_MCHBAR)—Offset 7220h

Config TDP level n. There are 3 instances of this register for 3 different levels.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 7220h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RO | **RSVD0:** Reserved |
| 62:48 | 0x0<br>RW | **PKG_MIN_POWER:** Min Power of SKU (units from PKG_POWER_SKU_UNIT) |
| 47 | 0x0<br>RO | **RSVD1:** Reserved |
| 46:32 | 0x0<br>RW | **PKG_MAX_POWER:** Max Power of SKU |
| 31:24 | 0x0<br>RO | **RSVD2:** Reserved |
| 23:16 | 0x0<br>RW | **TDP_RATIO:** TDP ratio for this level |
| 15 | 0x0<br>RO | **RSVD3:** Reserved |
| 14:0 | 0x0<br>RW | **TDP_POWER_VALUE:** TDP power value (Units from PKG_POWER_SKU_UNIT) |

## 44.3.210 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL2_0_0_0_MCHBAR)—Offset 7228h

Config TDP level n. There are 3 instances of this register for 3 different levels.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [MCHBAR] + 7228h

**Default:** 0000000000000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RO | **RSVD0:** Reserved |
| 62:48 | 0x0<br>RW | **PKG_MIN_POWER:** Min Power of SKU (units from PKG_POWER_SKU_UNIT) |
| 47 | 0x0<br>RO | **RSVD1:** Reserved |
| 46:32 | 0x0<br>RW | **PKG_MAX_POWER:** Max Power of SKU |
| 31:24 | 0x0<br>RO | **RSVD2:** Reserved |
| 23:16 | 0x0<br>RW | **TDP_RATIO:** TDP ratio for this level |
| 15 | 0x0<br>RO | **RSVD3:** Reserved |
| 14:0 | 0x0<br>RW | **TDP_POWER_VALUE:** TDP power value (Units from PKG_POWER_SKU_UNIT) |

## 44.3.211 CONFIG_TDP_LEVEL (P_CR_CONFIG_TDP_LEVEL3_0_0_0_MCHBAR)—Offset 7230h

Config TDP level n. There are 3 instances of this register for 3 different levels.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MCHBAR] + 7230h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RO | **RSVD0:** Reserved |
| 62:48 | 0x0<br>RW | **PKG_MIN_POWER:** Min Power of SKU (units from PKG_POWER_SKU_UNIT) |
| 47 | 0x0<br>RO | **RSVD1:** Reserved |
| 46:32 | 0x0<br>RW | **PKG_MAX_POWER:** Max Power of SKU |
| 31:24 | 0x0<br>RO | **RSVD2:** Reserved |
| 23:16 | 0x0<br>RW | **TDP_RATIO:** TDP ratio for this level |
| 15 | 0x0<br>RO | **RSVD3:** Reserved |
| 14:0 | 0x0<br>RW | **TDP_POWER_VALUE:** TDP power value (Units from PKG_POWER_SKU_UNIT) |

## 44.3.212 CONFIG_TDP_CONTROL (P_CR_CONFIG_TDP_CONTROL_0_0_0_MCHBAR)—Offset 7238h

Config TDP control

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MCHBAR] + 7238h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **CONFIG_TDP_LOCK:** Lock this register. BIOS should set this after selecting ConfigTDP level |
| 30:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1:0 | 0x0<br>RW/L | **CONFIG_TDP_LEVEL:** Used to select which configTDP level is desired |

## 44.3.213 CONFIG_TDP_NOMINAL (P_CR_CONFIG_TDP_NOMINAL_0_0_0_MCHBAR)—Offset 723Ch

Config TDP nominal

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 723Ch

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RW | **TDP_RATIO:** Config TDP ratio |

## 44.3.214 Graphics Superqueue Active Clocks (P_CR_PP1_ANY_THREAD_ACTIVITY_0_0_0_MCHBAR)—Offset 7244h

Graphics Superqueue active residency counter. Counts at ART)) 4.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MCHBAR] + 7244h

**Default:** 00000000h

**MCHBAR Type:** PCI Configuration Register (Size: 64 bits)
**MCHBAR Reference:** [B:0, D:0, F:0] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **SUPERQUEUE_ACTIVE_RESIDENCY:** Graphics Superqueue active residency counter. Counts in 19.2MHz reference clocks. |

## 44.3.215 B-Unit Copy of Default VTd BAR PMEN (B_CR_PMEN_REG_0_0_0_DEFVTDBAR)—Offset 64h

B-Unit copy of the Default VTd BAR PMEN register. Bit 31 is the only bit used for B-Unit.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 64h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** MCHBAR + 6C80)h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/V | **EPM:** This field controls DMA accesses to the protected lowmemory and protected highmemory regions.<br>• 0: Protected memory regions are disabled.<br>• 1: Protected memory regions are enabled. |
| 30:1 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |
| 0 | 0x0<br>RO | **PRS Unused (PRS):** Unused by the B-Unit. This field indicates the status of protected memory regions:<br>• 0: Protected memory regions are disabled.<br>• 1: Protected memory regions are enabled. |

## 44.3.216 B-Unit Copy of Default VTd BAR PLM Base Register (B_CR_PLMBASE_REG_0_0_0_DEFVTDBAR)—Offset 68h

Register to set up the base address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding the most significant bit position with 0 in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software must setup the protected low memory region below 4GB. Section 10.4.18 of the Intel Virtualization Technology for Directed I/O: Spec describes the Protected Low-Memory Limit register and hardware decoding of these registers. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 68h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** MCHBAR + 6C80)h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW/V | **Protected Low Memory Base (PLMB):** This register specifies the base of the protected low-memory region in system memory. |
| 19:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

## 44.3.217 B-Unit Copy of Default VTd BAR PLM Limit Register (B_CR_PLMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 6Ch

This register sets up the limit address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and it must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and then finding most significant zero bit position with 0 in the value read back from the register. Bits N:0 of the limit register are decoded by hardware as all 1s.

The Protected low-memory base and limit registers function as follows:

Programming the protected low-memory base and limit registers with the same value in bits 31:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. Programming the protected low-memory limit register with a value less than the protected low-memory base register disables the protected low-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [DEFVTDBAR] + 6Ch

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** MCHBAR + 6C80)h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW/V | **Protected Low Memory Limit (PLML):** This register specifies the last host physical address of the DMAprotected lowmemory region in system memory. |
| 19:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

## 44.3.218 B-Unit Copy of Default VTd BAR PHM Base Register (B_CR_PHMBASE_REG_0_0_0_DEFVTDBAR)—Offset 70h

This is the register to set up the base address of the DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software may setup the protected high memory region either above or below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 70h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** MCHBAR + 6C80)h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:39 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 38:20 | 0x0<br>RW/V | **Protected High Memory Base (PHMB):** This register specifies the base of protected high memory region in system memory.<br>Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width. |
| 19:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

## 44.3.219 B-Unit Copy of Default VTd BAR PHM Limit Register (B_CR_PHMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 78h

Register to set up the limit address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine the value of N by writing all ones to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of the limit register are decoded by hardware as all ones.

The protected high-memory base and limit registers function as follows:

Programming the protected low-memory base and limit registers with the same value in bits HAW:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. Programming the protected high-memory limit register with a value less than the protected high-memory base register disables the protected high-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 78h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** MCHBAR + 6C80)h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:39 | 0x0<br>RO | **Reserved (RESERVED_1):** Reserved. |
| 38:20 | 0x0<br>RW/V | **Protected High Memory Limit (PHML):** This register specifies the last host physical address of the DMA protected high memory region in system memory.<br>Hardware ignores and does not implement bits 63:HAW where HAW is the host address width. |
| 19:0 | 0x0<br>RO | **Reserved (RESERVED_0):** Reserved. |

# 44.4 Registers in Memory Space—DEFVTDBAR

## 44.4.1 Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h

Register to report the architecture version supported. Backward compatibility for the architecture is maintained with new revision numbers, allowing software to load remapping hardware drivers written for prior architecture versions.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 0h

**Default:** 00000010h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:4 | 0x1<br>RO | **Major Version Number (MAJOR):** Indicates supported architecture version. |
| 3:0 | 0x0<br>RO | **Minor Version Number (MINOR):** Indicates supported architecture minor version. |

## 44.4.2 Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h

Register to report general remapping hardware capabilities.

Note: These values can change based on defeature bits.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 8h

**Default:** 00D2008C40660462h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:57 | 0x0<br>RO | **Reserved:** Reserved |
| 56 | 0x0<br>RO | **First Level 64-KByte Page SupportP (FL1GP):** A value of 1 in this field indicates 1-GByte page size is supported for first-level translation. |
| 55 | 0x1<br>RO | **Read Draining (DRD):**<br>• 0 = Hardware does not support draining of DMA read requests.<br>• 1 = Hardware supports draining of DMA read requests. |
| 54 | 0x1<br>RO | **Write Draining (DWD):**<br>• 0 = Hardware does not support draining of DMA write requests.<br>• 1 = Hardware supports draining of DMA write requests. |
| 53:48 | 0x12<br>RO | **Reserved:** Reserved |
| 47:40 | 0x0<br>RO | **Reserved:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 39 | 0x1<br>RO | **Reserved:** Reserved |
| 38 | 0x0<br>RO | **Reserved:** Reserved |
| 37:34 | 0x3<br>RO | **Second Level Large Page Support (SLLPS):** This field indicates the super page sizes supported by hardware.<br>A 3 = 48-bit offset to page frame (1TB) |
| 33:24 | 0x40<br>RO | **Reserved:** Reserved |
| 23 | 0x0<br>RO | **Reserved:** Reserved |
| 22 | 0x1<br>RO | **Reserved:** Reserved |
| 21:16 | 0x26<br>RO | **Reserved:** Reserved |
| 15:13 | 0x0<br>RO | **Reserved:** Reserved |
| 12:8 | 0x4<br>RO | **Reserved:** Reserved |
| 7 | 0x0<br>RO | **Caching Mode (CM):**<br>• 0 = Not-present and erroneous entries are not cached in any of the remapping caches. Invalidations are not required for modifications to individual not present or invalid entries. However, any modifications that result in decreasing the effective permissions or partial permission increases require invalidations for them to be effective.<br>• 1 = Not-present and erroneous mappings may be cached in the remapping caches. Any software updates to the remapping structures (including updates to not-present or erroneous entries) require explicit invalidation. |
| 6 | 0x1<br>RO | **Protected High-Memory Region (PHMR):**<br>• 0 = Indicates protected high-memory region is not supported.<br>• 1 = Indicates protected high-memory region is supported. |
| 5 | 0x1<br>RO | **Protected Low-Memory Region (PLMR):**<br>• 0 = Indicates protected low-memory region is not supported.<br>• 1 = Indicates protected low-memory region is supported. |
| 4 | 0x0<br>RO | **Required Write-Buffer Flushing (RWBF):**<br>• 0 = Indicates no write-buffer flushing is needed to ensure changes to memory-resident structures are visible to hardware.<br>• 1 = Indicates software must explicitly flush the write buffers to ensure updates made to memory-resident remapping structures are visible to hardware. |
| 3 | 0x0<br>RO | **Advanced Fault Logging (AFL):**<br>• 0 = Indicates advanced fault logging is not supported. Only primary fault logging is supported.<br>• 1 = Indicates advanced fault logging is supported. |
| 2:0 | 0x2<br>RO | **Number of Domains Supported (ND):**<br>• 010b = Hardware supports 8-bit domain-ids with support for up to 256 domains. |

## 44.4.3 Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h

Register to report remapping hardware extended capabilities. Note: These values can change based on defeature bits.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 10h

**Default:** 0000000000F050DAh

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:40 | 0x0<br>RO | **RSVD0:** Reserved |
| 39:35 | 0x0<br>RO | **PASID Size Supported (PSS):** This field reports the PASID size supported by the remapping hardware for requests-with-PASID.<br>A value of N in this field indicates hardware supports PASID field of N+1 bits (For example, value of 7 in this field, indicates 8-bit PASIDs are supported).<br>Requests-with-PASID with PASID value beyond the limit specified by this field are treated as error by the remapping hardware.<br>This field is valid only when PASID field is reported as Set. |
| 34 | 0x0<br>RO | **Extended Accessed Flag Support (EAFS):**<br>• 0 = Hardware does not support the extended-accessed (EA) bit in first-level paging-structure entries.<br>• 1 = Hardware supports the extended accessed (EA) bit in first-level paging-structure entries.<br>This field is valid only when PASID field is reported as Set. |
| 33 | 0x0<br>RO | **No Write Flag Support (NWFS):**<br>• 0 = Hardware ignores the No Write (NW) flag in Device-TLB translation requests, and behaves as if NW is always 0.<br>• 1 = Hardware supports the No Write (NW) flag in Device-TLB translation requests.<br>This field is valid only when Device-TLB support (DT) field is reported as Set. |
| 32 | 0x0<br>RO | **PASID-Only Translations (POT):**<br>• 0 = Hardware does not support PASID-only Translation Type in extended-context-entries.<br>• 1 = Hardware supports PASID-only Translation Type in extended-context-entries.<br>This field is valid only when PASID field is reported as Set. |
| 31 | 0x0<br>RO | **Supervisor Request Support (SRS):**<br>• 0 = H/W does not support requests-with-PASID seeking supervisor privilege.<br>• 1 = H/W supports requests-with-PASID seeking supervisor privilege.<br>The field is valid only when PASID field is reported as Set. |
| 30 | 0x0<br>RO | **Execute Request Support (ERS):**<br>• 0 = H/W does not support requests-with-PASID seeking execute permission.<br>• 1 = H/W supports requests-with-PASID seeking execute permission.<br>This field is valid only when PASID field is reported as Set. |
| 29 | 0x0<br>RO | **Page Request Support (PRS):**<br>• 0 = Hardware does not support Page Requests.<br>• 1 = Hardware supports Page Requests<br>This field is valid only when Device-TLB (DT) field is reported as Set. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 28 | 0x0<br>RO | **Process Address Space ID Support (PASID):**<br>• 0 = Hardware does not support requests tagged with Process Address Space IDs.<br>• 1 = Hardware supports requests tagged with Process Address Space IDs. |
| 27 | 0x0<br>RO | **Deferred Invalidate Support (DIS):**<br>• 0 = Hardware does not support deferred invalidations of IOTLB and Device-TLB.<br>• 1 = Hardware supports deferred invalidations of IOTLB and Device-TLB.<br>This field is valid only when PASID field is reported as Set. |
| 26 | 0x0<br>RO | **Nested Translation Support (NEST):**<br>• 0 = Hardware does not support nested translations.<br>• 1 = Hardware supports nested translations.<br>This field is valid only when PASID field is reported as Set. |
| 25 | 0x0<br>RO | **Memory Type Support (MTS):**<br>• 0 = Hardware does not support Memory Type in first-level translation and Extended Memory type in second-level translation.<br>• 1 = Hardware supports Memory Type in first-level translation and Extended Memory type in second-level translation.<br>This field is valid only when PASID and ECS fields are reported as Set.<br>Remapping hardware units with, one or more devices that operate in processor coherency domain, under its scope must report this field as Set. |
| 24 | 0x0<br>RO | **Extended Context Support (ECS):**<br>• 0 = Hardware does not support extended-root-entries and extended-context-entries.<br>• 1 = Hardware supports extended-root-entries and extended-context-entries.<br>Implementations reporting PASID or PRS fields as Set, must report this field as Set. |
| 23:20 | 0xf<br>RO | **Maximum Handle Mask Value (MHMV):** The value in this field indicates the maximum supported value for the Handle Mask (HM) field in the interrupt entry cache invalidation descriptor (iec_inv_dsc). This field is valid only when the IR field in Extended Capability register is reported as Set. |
| 19:18 | 0x0<br>RO | **RSVD1:** Reserved |
| 17:8 | 0x50<br>RO | **IOTLB Register Offset (IRO):** This field specifies the offset to the IOTLB registers relative to the register base address of this remapping hardware unit.<br>If the register base address is X, and the value reported in this field is Y, the address for the first IOTLB invalidation register is calculated as X+(16*Y). |
| 7 | 0x1<br>RO | **Snoop Control (SC):**<br>• 0 = Hardware does not support 1-setting of the SNP field in the page-table entries.<br>• 1 = Hardware supports the 1-setting of the SNP field in the page-table entries. |
| 6 | 0x1<br>RO | **Pass Through (PT):**<br>• 0 = Hardware does not support pass-through translation type in context entries and extended-context-entries.<br>• 1 = Hardware supports pass-through translation type in context entries and extended-context-entries.<br>Pass-through translation is specified through Translation-Type (T) field value of 10b in context-entries, or T field value of 010b in extended-context-entries.<br>Hardware implementations supporting PASID must report a value of 1b in this field. |
| 5 | 0x0<br>RO | **RSVD2:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x1<br>RO | **Extended Interrupt Mode (EIM):**<br>• 0 = On Intel64 platforms, hardware supports only 8-bit APIC-IDs (xAPIC mode).<br>• 1 = On Intel64 platforms, hardware supports 32-bit APIC-IDs (x2APIC mode).<br>This field is valid only on Intel64 platforms reporting Interrupt Remapping support (IR field Set). |
| 3 | 0x1<br>RO | **Interrupt Remapping Suport (IR):**<br>• 0 = Hardware does not support interrupt remapping.<br>• 1 = Hardware supports interrupt remapping.<br>Implementations reporting this field as Set must also support Queued Invalidation (QI). |
| 2 | 0x0<br>RO | **Device-TLB Support (DT):**<br>• 0 = Hardware does not support device-IOTLBs.<br>• 1 = Hardware supports Device-IOTLBs.<br>Implementations reporting this field as Set must also support Queued Invalidation (QI). Hardware implementations supporting I/O Page Requests (PRS field Set in Extended Capability register) must report a value of 1b in this field. |
| 1 | 0x1<br>RO | **Queued Invalidation Support (QI):**<br>• 0 = Hardware does not support queued invalidations.<br>• 1 = Hardware supports queued invalidations. |
| 0 | 0x0<br>RO | **Page-Walk Coherency (C):** This field indicates if hardware access to the root, context, extended-context and interrupt-remap tables, and second-level paging structures for requests-without-PASID, are coherent (snooped) or not.<br>• 0 = Indicates hardware accesses to remapping structures are non-coherent.<br>• 1 = Indicates hardware accesses to remapping structures are coherent.<br>Hardware access to advanced fault log, invalidation queue, invalidation semaphore, page-request queue, PASID-table, PASID-state table, and first-level page-tables are always coherent. |

## 44.4.4 Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h

Register to control remapping hardware. If multiple control fields in this register need to be modified, software must serialize the modifications through multiple writes to this register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 18h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Translation Enable (TE):** Software writes to this field to request hardware to enable/disable DMA-remapping:<br>• 0 = Disable DMA remapping.<br>• 1 = Enable DMA remapping. |
| 30 | 0x0<br>WO | **Set Root Table Pointer (SRTP):** Software sets this field to set/update the root-entry table pointer used by hardware.<br>The root-entry table pointer is specified through the Root-entry Table Address (RTA_REG) register.<br>The Set Root Table Pointer operation must be performed before enabling or re-enabling (after disabling) DMA remapping through the TE field. |
| 29 | 0x0<br>RO | **Reserved:** Reserved |
| 28 | 0x0<br>RO | **Reserved:** Reserved |
| 27 | 0x0<br>RO | **Reserved:** Reserved |
| 26 | 0x0<br>RW | **Queued Invalidation Enable (QIE):** This field is valid only for implementations supporting queued invalidations. Software writes to this field to enable or disable queued invalidations.<br>• 0 = Disable queued invalidations.<br>• 1 = Enable use of queued invalidations. |
| 25 | 0x0<br>RW | **Interrupt Remapping Enable (IRE):** This field is valid only for implementations supporting interrupt remapping.<br>• 0 = Disable interrupt-remapping hardware.<br>• 1 = Enable interrupt-remapping hardware. |
| 24 | 0x0<br>WO | **Set Interrupt Remap Table Pointer (SIRTP):** This field is valid only for implementations supporting interrupt-remapping.<br>Software sets this field to set/update the interrupt remapping table pointer used by hardware.<br>The interrupt remapping table pointer is specified through the Interrupt Remapping Table Address (IRTA_REG) register. |
| 23 | 0x0<br>RW | **Compatibility Format Interrupt (CFI):** This field is valid only for Intel64 implementations supporting interrupt-remapping. Software writes to this field to enable or disable Compatibility Format interrupts on Intel64 platforms. The value in this field is effective only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled.<br>• 0 = Block Compatibility format interrupts.<br>• 1 = Process Compatibility format interrupts as pass-through (bypass interrupt remapping). |
| 22:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.5 Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch

Register to report general remapping hardware status.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 1Ch

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Reserved:** Reserved |
| 30 | 0x0<br>RO/V | **Reserved:** Reserved |
| 29 | 0x0<br>RO | **Fault Log Status (FLS):** This field:<br>• Is cleared by hardware when software Sets the SFL field in the Global Command register.<br>• Is Set by hardware when hardware completes the Set Fault Log Pointer operation using the value provided in the Advanced Fault Log register. |
| 28 | 0x0<br>RO | **Advanced Fault Logging Status (AFLS):** This field is valid only for implementations supporting advanced fault logging. It indicates the advanced fault logging status:<br>• 0 = Advanced Fault Logging is not enabled.<br>• 1 = Advanced Fault Logging is enabled. |
| 27 | 0x0<br>RO | **Write Buffer Flush Status (WBFS):** This field is valid only for implementations requiring write buffer flushing. This field indicates the status of the write buffer flush command. |
| 26 | 0x0<br>RO/V | **Queued Invalidation Enable Status (QIES):** This field indicates queued invalidation enable status.<br>• 0 = queued invalidation is not enabled.<br>• 1 = queued invalidation is enabled |
| 25 | 0x0<br>RO/V | **Interrupt Remapping Enable Status (IRES):** This field indicates the status of Interrupt-remapping hardware.<br>• 0 = Interrupt-remapping hardware is not enabled.<br>• 1 = Interrupt-remapping hardware is enabled |
| 24 | 0x0<br>RO/V | **Interrupt Remapping Pointer Status (IRTPS):** This field indicates the status of the interrupt remapping table pointer in hardware.<br>This field is cleared by hardware when software sets the SIRTP field in the Global Command register. |
| 23 | 0x0<br>RO/V | **Compatibility Format Interrupt Status (CFIS):** This field indicates the status of Compatibility format interrupts on Intel64 implementations supporting interrupt-remapping.<br>The value reported in this field is applicable only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled.<br>• 0 = Compatibility format interrupts are blocked.<br>• 1 = Compatibility format interrupts are processed as pass-through (bypassing interrupt remapping). |
| 22:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.6 Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h

Register providing the base address of root-entry table.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 20h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:12 | 0x0<br>RW | **Root Table Address (RTA):** This register points to base of page aligned, 4KB-sized root-entry table in system memory. |
| 11 | 0x0<br>RW | **Root Table Type (RTT):** This field specifies the type of root-table referenced by the Root Table Address (RTA) field:<br>• 0 = Root Table.<br>• 1 = Extended Root Table |
| 10:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.7 Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h

Register to manage context cache. The act of writing the uppermost byte of the CCMD_REG with the ICC field Set causes the hardware to perform the context-cache invalidation.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 28h

**Default:** 0800000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/V | **Invalidate Context Cache (ICC):** Software requests invalidation of context-cache by setting this field.<br>Software must also set the requested invalidation granularity by programming the CIRG field.<br>Software must read back and check the ICC field is Clear to confirm the invalidation is complete.<br>Software must not update this register when this field is set.<br>Hardware clears the ICC field to indicate the invalidation request is complete.<br>Hardware also indicates the granularity at which the invalidation operation was performed through the CAIG field.<br>Software must submit a context-cache invalidation request through this field only when there are no invalidation requests pending at this remapping hardware unit.<br>Since information from the context-cache may be used by hardware to tag IOTLB entries, software must perform domain-selective (or global) invalidation of IOTLB after the context cache invalidation has completed.<br>Hardware implementations reporting write-buffer flushing requirement (RWBF=1 in Capability register) must implicitly perform a write buffer flush before invalidating the context cache. |
| 62:61 | 0x0<br>RW | **Context Invalidation Request Granularity (CIRG):** Software provides the requested invalidation granularity through this field when setting the ICC field:<br>• 00: Reserved.<br>• 01: Global Invalidation request.<br>• 10: Domain-selective invalidation request. The target domain-id must be specified in the DID field.<br>• 11: Device-selective invalidation request. The target source-id(s) must be specified through the SID and FM fields, and the domain-id (that was programmed in the context-entry for these device(s)) must be provided in the DID field.<br>Hardware implementations may process an invalidation request by performing invalidation at a coarser granularity than requested.<br>Hardware indicates completion of the invalidation request by clearing the ICC field.<br>At this time, hardware also indicates the granularity at which the actual invalidation was performed through the CAIG field. |
| 60:59 | 0x1<br>RO/V | **Context Actual Invalidation Granularity (CAIG):** Hardware reports the granularity at which an invalidation request was processed through the CAIG field at the time of reporting invalidation completion (by clearing the ICC field).<br>The following are the encodings for this field:<br>• 00: Reserved.<br>• 01: Global Invalidation performed. This could be in response to a global, domain-selective or device-selective invalidation request.<br>• 10: Domain-selective invalidation performed using the domain-id specified by software in the DID field. This could be in response to a domain-selective or device-selective invalidation request.<br>• 11: Device-selective invalidation performed using the source-id and domain-id specified by software in the SID and FM fields. This can only be in response to a device-selective invalidation request. |
| 58:34 | 0x0<br>RO | **RSVD0:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 33:32 | 0x0<br>RW | **Function Mask (FM):** Software may use the Function Mask to perform device-selective invalidations on behalf of devices supporting PCI Express Phantom Functions.<br>This field specifies which bits of the function number portion (least significant three bits) of the SID field to mask when performing device-selective invalidations.<br>The following encodings are defined for this field:<br>• 00: No bits in the SID field masked.<br>• 01: Mask most significant bit of function number in the SID field.<br>• 10: Mask two most significant bit of function number in the SID field.<br>• 11: Mask all three bits of function number in the SID field.<br>The context-entries corresponding to all the source-ids specified through the FM and SID fields must have to the domain-id specified in the DID field. |
| 31:16 | 0x0<br>RW | **Source-ID (SID):** Indicates the source-id of the device whose corresponding context-entry needs to be selectively invalidated.<br>This field along with the FM field must be programmed by software for device-selective invalidation requests. |
| 15:0 | 0x0<br>RW | **Domain-ID (DID):** Indicates the id of the domain whose context-entries need to be selectively invalidated.<br>This field must be programmed by software for both domain-selective and device-selective invalidation requests.<br>The Capability register reports the domain-id width supported by hardware.<br>Software must ensure that the value written to this field is within this limit.<br>Hardware may ignore and not implement bits15:N, where N is the supported domain-id width reported in the Capability register. |

## 44.4.8 Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h

Register indicating the various error status.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 34h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:8 | 0x0<br>RO | **Fault Record Index (FRI):** This field is valid only when the PPF field is Set.<br>The FRI field indicates the index (from base) of the fault recording register to which the first pending fault was recorded when the PPF field was Set by hardware.<br>The value read from this field is undefined when the PPF field is clear. |
| 7 | 0x0<br>RW/1C | **Page Request Overflow (PRO):** Hardware detected a Page Request Overflow error.<br>Hardware implementations not supporting the Page Request Queue implement this bit as RsvdZ.<br><br>**Power Well:** prst |
| 6 | 0x0<br>RO | **Invalidation Time-out Error (ITE):** Hardware detected a Device-IOTLB invalidation completion time-out.<br>At this time, a fault event may be generated based on the programming of the Fault Event Control register.<br>Hardware implementations not supporting device Device-IOTLBs implement this bit as RsvdZ. |
| 5 | 0x0<br>RO | **Invalidation Completion Error (ICE):** Hardware received an unexpected or invalid Device-IOTLB invalidation completion.<br>This could be due to either an invalid ITag or invalid source-id in an invalidation completion response.<br>At this time, a fault event may be generated based on the programming of hte Fault Event Control register.<br>Hardware implementations not supporting Device-IOTLBs implement this bit as RsvdZ. |
| 4 | 0x0<br>RW/1C | **Invalidation Queue Error (IQE):** Hardware detected an error associated with the invalidation queue.<br>This could be due to either a hardware error while fetching a descriptor from the invalidation queue, or hardware detecting an erroneous or invalid descriptor in the invalidation queue.<br>At this time, a fault event may be generated based on the programming of the Fault Event Control register.<br>Hardware implementations not supporting queued invalidations implement this bit as RsvdZ.<br><br>**Power Well:** prst |
| 3 | 0x0<br>RO | **Advanced Pending Fault (APF):** When this field is Clear, hardware sets this field when the first fault record (at index 0) is written to a fault log.<br>At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it.<br>Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ. |
| 2 | 0x0<br>RO | **Advanced Fault Overflow (AFO):** Hardware sets this field to indicate advanced fault log overflow condition.<br>At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it.<br>Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ. |
| 1 | 0x0<br>RO/V | **Primary Pending Fault (PPF):** This field indicates if there are one or more pending faults logged in the fault recording registers.<br>Hardware computes this field as the logical OR of Fault (F) fields across all the fault recording registers of this remapping hardware unit.<br>• 0 = No pending faults in any of the fault recording registers.<br>• 1 = One or more fault recording registers has pending faults. The FRI field is updated by hardware whenever the PPF field is set by hardware. Also, depending on the programming of Fault Event Control register, a fault event is generated when hardware sets this field.<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RW/1C | **Primary Fault Overflow (PFO):** Hardware sets this field to indicate overflow of fault recording registers.<br>Software writing 1 clears this field.<br>When this field is Set, hardware does not record any new faults until software clears this field.<br><br>**Power Well:** prst |

## 44.4.9 Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h

Register specifying the fault event interrupt message control bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 38h

**Default:** 80000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Interrupt Mask (IM):**<br>• 0 = No masking of interrupt. When an interrupt condition is detected, hardware issues an interrupt message (using the Fault Event Data and Fault Event Address register values).<br>• 1 = This is the value on reset. Software may mask interrupt message generation by setting this field. Hardware is prohibited from sending the interrupt message when this field is set. |
| 30 | 0x0<br>RO/V | **Interrupt Pending (IP):** Hardware sets the IP field whenever it detects an interrupt condition, which is defined as:<br>• When primary fault logging is active, an interrupt condition occurs when hardware records a fault through one of the Fault Recording registers and sets the PPF field in Fault Status register.<br>• When advanced fault logging is active, an interrupt condition occurs when hardware records a fault in the first fault record (at index 0) of the current fault log and sets the APF field in the Fault Status register.<br>• Hardware detected error associated with the Invalidation Queue, setting the IQE field in the Fault Status register.<br>• Hardware detected invalid Device-IOTLB invalidation completion, setting the ICE field in the Fault Status register.<br>• Hardware detected Device-IOTLB invalidation completion time-out, setting the ITE field in the Fault Status register.<br>If any of the status fields in the Fault Status register was already Set at the time of setting any of these fields, it is not treated as a new interrupt condition.<br>The IP field is kept set by hardware while the interrupt message is held pending.<br>The interrupt message could be held pending due to interrupt mask (IM field) being Set or other transient hardware conditions.<br>The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either:<br>• Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending, or due to software clearing the IM field.<br>• Software servicing all the pending interrupt status fields in the Fault Status register as follows:<br>• When primary fault logging is active, software clearing the Fault (F) field in all the Fault Recording registers with faults, causing the PPF field in Fault Status register to be evaluated as clear.<br>• Software clearing other status fields in the Fault Status register by writing back the value read from the respective fields. |
| 29:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.10 Fault Event Data Register (FEDATA_REG_0_0_0_VTDBAR)—Offset 3Ch

Register specifying the interrupt message data

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 3Ch

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Extended Interrupt Message Data (EIMD):** This field is valid only for implementations supporting 32-bit interrupt data fields.<br>Hardware implementations supporting only 16-bit interrupt data may treat this field as RsvdZ. |
| 15:0 | 0x0<br>RW | **Interrupt Message Data (IMD):** Data value in the interrupt request. |

## 44.4.11 Fault Event Address Register (FEADDR_REG_0_0_0_VTDBAR)—Offset 40h

Register specifying the interrupt message address.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 40h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** When fault events are enabled, the contents of this register specify the DWORD-aligned address (bits 31:2) for the interrupt request. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.12 Fault Event Upper Address Register (FEUADDR_REG_0_0_0_VTDBAR)—Offset 44h

Register specifying the interrupt message upper address.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 44h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Hardware implementations supporting Extended Interrupt Mode are required to implement this register.<br>Hardware implementations not supporting Extended Interrupt Mode may treat this field as RsvdZ. |

## 44.4.13 Advanced Fault Log Register (AFLOG_REG_0_0_0_VTDBAR)—Offset 58h

Register to specify the base address of the memory-resident fault-log region. This register is treated as RsvdZ for implementations not supporting advanced translation fault logging (AFL field reported as 0 in the Capability register).

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 58h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:12 | 0x0<br>RO | **Fault Log Address (FLA):** This field specifies the base of 4KB aligned fault-log region in system memory.<br>Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width. Software specifies the base address and size of the fault log region through this register, and programs it in hardware through the SFL field in the Global Command register.<br>When implemented, reads of this field return the value that was last programmed to it. |
| 11:9 | 0x0<br>RO | **Fault Log Size (FLS):** This field specifies the size of the fault log region pointed by the FLA field.<br>The size of the fault log region is 2X * 4KB, where X is the value programmed in this register.<br>When implemented, reads of this field return the value that was last programmed to it. |
| 8:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.14 Protected Memory Enable Register (PMEN_REG_0_0_0_VTDBAR)—Offset 64h

Register to enable the DMA-protected memory regions setup through the PLMBASE,..PLMLIMT, PHMBASE, PHMLIMIT registers. This register is always treated as RO for implementations not supporting protected memory regions (PLMR and PHMR fields reported as Clear in the Capability register). Protected memory regions may be used by software to securely initialize remapping structures in memory. To avoid impact to legacy BIOS usage of memory, software is recommended to not overlap protected memory regions with any reserved memory regions of the platform reported through the Reserved Memory Region Reporting (RMRR) structures.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 64h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Enable Protected Memory (EPM):** This field controls DMA accesses to the protected low-memory and protected high-memory regions.<br>• 0 = Protected memory regions are disabled.<br>• 1 = Protected memory regions are enabled.DMA requests accessing protected memory regions are handled as follows:<br>When DMA remapping is not enabled, all DMA requests accessing protected memory regions are blocked.<br>When DMA remapping is enabled:<br>DMA requests processed as pass-through (Translation Type value of 10b in Context-Entry) and accessing the protected memory regions are blocked.<br>DMA requests with translated address (AT=10b) and accessing the protected memory regions are blocked.<br>DMA requests that are subject to address remapping, and accessing the protected memory regions may or may not be blocked by hardware. For such requests, software must not depend on hardware protection of the protected memory regions, and instead program the DMA-remapping page-tables to not allow DMA to protected memory regions.<br>Remapping hardware access to the remapping structures are not subject to protected memory region checks. DMA requests blocked due to protected memory region violation are not recorded or reported as remapping faults.<br>Hardware reports the status of the protected memory enable/disable operation through the PRS field in this register.<br>Hardware implementations supporting DMA draining must drain any in-flight translated DMA requests queued within the Root-Complex before indicating the protected memory region as enabled through the PRS field. |
| 30:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RO/V | **Protected Region Status (PRS):** This field indicates the status of protected memory region(s):<br>• 0 = Protected memory region(s) disabled.<br>• 1 = Protected memory region(s) enabled. |

## 44.4.15 Protected Low Memory Base Register (PLMBASE_REG_0_0_0_VTDBAR)—Offset 68h

Register to set up the base address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding the most significant zero bit position with 0 in the value read back from the register. Bits N:0 of this register is decoded by hardware as all 0s...Software must setup the protected low memory region below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 68h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Protected Low-Memory Base (PLMB):** This register specifies the base of protected low-memory region in system memory. |
| 19:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.16 Protected Low-Memory Limit Register (PLMLIMIT_REG_0_0_0_VTDBAR)—Offset 6Ch

Register to set up the limit address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register) The alignment of the protected low memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding most significant zero bit position with 0 in the value read back from the register. Bits N:0 of the limit register is decoded by hardware as all 1s The Protected low-memory base and limit registers functions as follows:

- Programming the protected low-memory base and limit registers with the same value in bits 31: (N+1) specifies a protected low-memory region of size 2(N+1) bytes
- Programming the protected low-memory limit register with a value less than the protected low-memory base register disables the protected low-memory region

Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 6Ch

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Protected Low-Memory Limit (PLML):** This register specifies the last host physical address of the DMA-protected low-memory region in system memory. |
| 19:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.17 Protected High-Memory Base Register (PHMBASE_REG_0_0_0_VTDBAR)—Offset 70h

Register to set up the base address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register) The alignment of the protected high memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s Software may setup the protected high memory region either above or below 4GB Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 70h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:39 | 0x0<br>RO | **RSVD0:** Reserved |
| 38:20 | 0x0<br>RW | **Protected High-Memory Base (PHMB):** This register specifies the base of protected (high) memory region in system memory.<br>Hardware ignores, and does not implement, bits 63:HAW, where HAW is the host address width. |
| 19:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 44.4.18 Protected High-Memory Limit Register (PHMLIMIT_REG_0_0_0_VTDBAR)—Offset 78h

Register to set up the limit address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register) The alignment of the protected high memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine the value of N by writing all 1s to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of the limit register is decoded by hardware as all 1s The protected high-memory base and limit registers functions as follows

- Programming the protected low-memory base and limit registers with the same value in bits HAW:(N+1) specifies a protected low-memory region of size 2(N+1) bytes
- Programming the protected high-memory limit register with a value less than the protected high-memory base register disables the protected high-memory region

Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 78h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:39 | 0x0<br>RO | **RSVD0:** Reserved |
| 38:20 | 0x0<br>RW | **Protected High-Memory Limit (PHML):** This register specifies the last host physical address of the DMA-protected high-memory region in system memory.<br>Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width. |
| 19:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 44.4.19 Invalidation Queue Head Register (IQH_REG_0_0_0_VTDBAR)—Offset 80h

Register indicating the invalidation queue head. This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 80h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18:4 | 0x0<br>RO/V | **Queue Head (QH):** Specifies the offset (128-bit aligned) to the invalidation queue for the command that will be fetched next by hardware.<br>Hardware resets this field to 0 whenever the queued invalidation is disabled (QIES field Clear in the Global Status register). |
| 3:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 44.4.20 Invalidation Queue Tail Register (IQT_REG_0_0_0_VTDBAR)—Offset 88h

Register indicating the invalidation tail head. This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 88h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18:4 | 0x0<br>RW | **Queue Tail (QT):** Specifies the offset (128-bit aligned) to the invalidation queue for the command that will be written next by software. |
| 3:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 44.4.21 Invalidation Queue Address Register (IQA_REG_0_0_0_VTDBAR)—Offset 90h

Register to configure the base address and size of the invalidation queue. This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 90h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:39 | 0x0<br>RO | **RSVD0:** Reserved |
| 38:12 | 0x0<br>RW | **Invalidation Queue Base Address (IQA):** This field points to the base of 4KB aligned invalidation request queue.<br>Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.<br>Reads of this field return the value that was last programmed to it. |
| 11:3 | 0x0<br>RO | **RSVD1:** Reserved |
| 2:0 | 0x0<br>RW | **Queue Size (QS):** This field specifies the size of the invalidation request queue. A value of X in this field indicates an invalidation request queue of (2X) 4KB pages.<br>The number of entries in the invalidation queue is 2(X + 8). |

## 44.4.22 Invalidation Completion Status Register (ICS_REG_0_0_0_VTDBAR)—Offset 9Ch

Register to report completion status of invalidation wait descriptor with Interrupt Flag (IF) Set This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + 9Ch

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RW/1C | **Invalidation Wait Descriptor Complete (IWC):** Indicates completion of Invalidation Wait Descriptor with Interrupt Flag (IF) field Set. Hardware implementations not supporting queued invalidations implement this field as RsvdZ.<br><br>**Power Well:** prst |

## 44.4.23 Invalidation Event Control Register (IECTL_REG_0_0_0_VTDBAR)—Offset A0h

Register specifying the invalidation event interrupt control bits This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + A0h

**Default:** 80000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Interrupt Mask (IM):**<br>• 0= No masking of interrupt. When a invalidation event condition is detected, hardware issues an interrupt message (using the Invalidation Event Data and Invalidation Event Address register values)<br>• 1= This is the value on reset. Software may mask interrupt message generation by setting this field. Hardware is prohibited from sending the interrupt message when this field is Set. |
| 30 | 0x0<br>RO/V | **Interrupt Pending (IP):** Hardware sets the IP field whenever it detects an interrupt condition. Interrupt condition is defined as:<br>• An Invalidation Wait Descriptor with Interrupt Flag (IF) field Set completed, setting the IWC field in the Invalidation Completion Status register<br>• If the IWC field in the Invalidation Completion Status register was already Set at the time of setting this field, it is not treated as a new interrupt condition<br>The IP field is kept Set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being Set, or due to other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either:<br>• 0= Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending or due to software clearing the IM field<br>• 1= Software servicing the IWC field in the Invalidation Completion Status register. |
| 29:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.24 Invalidation Event Data Register (IEDATA_REG_0_0_0_VTDBAR)—Offset A4h

Register specifying the Invalidation Event interrupt message data This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + A4h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Extended Interrupt Message Data (EIMD):** This field is valid only for implementations supporting 32-bit interrupt data fields.<br>Hardware implementations supporting only 16-bit interrupt data treat this field as Rsvd. |
| 15:0 | 0x0<br>RW | **Interrupt Message Data (IMD):** Data value in the interrupt request. |

## 44.4.25 Invalidation Event Address Register (IEADDR_REG_0_0_0_VTDBAR)—Offset A8h

Register specifying the Invalidation Event Interrupt message address This register is treated as RsvdZ by implementations reporting Queued Invalidation (QI) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + A8h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** When fault events are enabled, the contents of this register specify the DWORD-aligned address (bits 31:2) for the interrupt request. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.26 Invalidation Event Upper Address Register (IEUADDR_REG_0_0_0_VTDBAR)—Offset ACh

Register specifying the Invalidation Event interrupt message upper address.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + ACh

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Hardware implementations supporting Queued Invalidations and Extended Interrupt Mode are required to implement this register.<br>Hardware implementations not supporting Queued Invalidations or Extended Interrupt Mode may treat this field as RsvdZ. |

## 44.4.27 Interrupt Remapping Table Address Register (IRTA_REG_0_0_0_VTDBAR)—Offset B8h

Register providing the base address of Interrupt remapping table.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + B8h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:12 | 0x0<br>RW | **Interrupt Remapping Table Address (IRTA):** This field points to the base of 4KB aligned interrupt remapping table |
| 11 | 0x0<br>RW | **Extended Interrupt Mode Enable (EIME):** This field is used by hardware on Intel64 platforms as follows:<br>• 0=xAPIC mode is active. Hardware interprets only low 8-bits of Destination-ID field in the IRTEs. The high 24-bits of the Destination-ID field are treated as reserved<br>• 1= x2APIC mode is active. Hardware interprets all 32-bits of Destination-ID field in the IRTEs |
| 10:4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3:0 | 0x0<br>RW | **Size (S):** This field specifies the size of the interrupt remapping table. The number of entries in the interrupt remapping table is 2(X+1), where X is the value programmed in this field. |

## 44.4.28 Page Request Status Register (PRESTS_REG_0_0_0_VTDBAR)—Offset DCh

Register to report pending page request in page request queue. This register is treated as RsvdZ by implementations reporting Page Request Support (PRS) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + DCh

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RW/1C | **Pending Page Request (PPR):** Pending Page Request: Indicates pending page requests to be serviced by software in the page request queue.<br>This field is Set by hardware when a streaming page request entry (page_stream_reg_dsc) or a page group request (page_grp_req_dsc) with Last Page in Group (LPG) field Set, is added to the page request queue.<br><br>**Power Well:** prst |

## 44.4.29 Page Request Event Control Register (PRECTL_REG_0_0_0_VTDBAR)—Offset E0h

Register specifying the page request event interrupt control bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + E0h

**Default:** 80000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Interrupt Mask (IM):** Interrupt Mask<br>• 0=No masking of interrupt. When a page request event condition is detected, hardware issues an interrupt message<br>• 1=This is the value on reset. Software may mask interrupt message generation by setting this field. |
| 30 | 0x0<br>RO/V | **Interrupt Pending (IP):** Interrupt Pending: Hardware sets the IP field whenever it detects an interrupt condition.<br>The IP field is kept Set by hardware while the interrupt message is held pending. |
| 29:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.30 Page Request Event Data Register (PREDATA_REG_0_0_0_VTDBAR)—Offset E4h

Register specifying the Page Request Event interrupt message data. This register is treated as RsvdZ by implementations reporting Page Request Support (PRS) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + E4h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Extended Interrupt Message Data (EIMD):** Extended Interrupt Message Data |
| 15:0 | 0x0<br>RW | **Interrupt Message Data (IMD):** Interrupt Message Data: Data value in the interrupt request. Software requirements for programming this register are described in VTd Spec |

## 44.4.31 Page Request Event Address Register (PREADDR_REG_0_0_0_VTDBAR)—Offset E8h

Register specifying the Page Request Event Interrupt message address. This register is treated as RsvdZ by implementations reporting Page Request Support (PRS) as not supported in the Extended Capability register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + E8h

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** Message Address: When fault events are enabled, the contents of this register specify the DWORD-aligned address (bits 31:2) for the interrupt request. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.32 Page Request Event Upper Address Register (PREUADDR_REG_0_0_0_VTDBAR)—Offset ECh

Register specifying the Page Request Event interrupt message upper address.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [DEFVTDBAR] + ECh

**Default:** 00000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Message Upper Address:<br>This field specifies the upper address (bits.. 63:32) for the page request event interrupt. |

## 44.4.33 Fault Recording Register Low [0] (FRCDL_REG_0_0_0_VTDBAR)—Offset 400h

Register to record fault information when primary fault logging is active. Hardware reports the number and location of fault recording registers through the Capability register. This register is relevant only for primary fault logging This register is sticky and can be cleared only through power good reset or by software clearing the RW1C fields by writing a 1.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 400h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:12 | 0x0<br>RO/V | **Fault Info (FI):** When the Fault Reason (FR) field indicates one of the DMA-remapping fault conditions, bits 63:12 of this field contain the page address in the faulted DMA request.<br>Hardware treats bits 63:N as reserved (0), where N is the maximum guest address width (MGAW) supported.<br>When the Fault Reason (FR) field indicates one of the interrupt-remapping fault conditions, bits 63:48 of this field indicate the interrupt_index computed for the faulted interrupt request, and bits 47:12 are cleared.<br>This field is relevant only when the F field is Set.<br><br>**Power Well:** prst |
| 11:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 44.4.34 Fault Recording Register High [0] (FRCDH_REG_0_0_0_VTDBAR)—Offset 408h

Register to record fault information when primary fault logging is active. Hardware reports the number and location of fault recording registers through the Capability register. This register is relevant only for primary fault logging This register is sticky and can be cleared only through power good reset or by software clearing the RW1C fields by writing a 1.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 408h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/1C | **Fault (F):** Hardware sets this field to indicate a fault is logged in this Fault Recording register.<br>The F field is set by hardware after the details of the fault is recorded in other fields.<br>When this field is Set, hardware may collapse additional faults from the same source-id (SID)<br><br>**Power Well:** prst |
| 62 | 0x0<br>RO/V | **Type (T):** Type of the faulted request:<br>• 0=0: Write request<br>• 1=1: Read request or AtomicOp request<br>This field is relevant only when the F field is Set, and when the fault reason (FR) indicates one of the DMA-remapping fault conditions.<br><br>**Power Well:** prst |
| 61:60 | 0x0<br>RO/V | **Address Type (AT):** This field captures the AT field from the faulted DMA request.<br>Hardware implementations not supporting Device-IOTLBs (DI field Clear in Extended Capability register) treat this field as RsvdZ.<br>When supported, this field is valid only when the F field is Set, and when the fault reason (FR) indicates one of the DMA-remapping fault conditions.<br><br>**Power Well:** prst |
| 59:40 | 0x0<br>RO/V | **PASID Value (PN):** PASID value in the faulted request.<br>This field is relevant only when the PP field is set. Hardware implementations not supporting PASID (PASID field Clear in Extended Capability register) implement this field as RsvdZ.<br><br>**Power Well:** prst |
| 39:32 | 0x0<br>RO/V | **Fault Reason (FR):** Reason for the fault<br>This field is relevant only when the F field is set.<br><br>**Power Well:** prst |
| 31 | 0x0<br>RO/V | **PASID Present (PP):** When set, indicates the faulted request has a PASID tag.<br>The value of the PASID field is reported in the PASID Value (PV) field.<br>This field is relevant only when the F field is Set, and when the fault reason (FR) indicates one of the non-recoverable address translation fault conditions. Hardware implementations not supporting PASID (PASID field Clear in Extended Capability register) implement this field as RsvdZ.<br><br>**Power Well:** prst |
| 30 | 0x0<br>RO/V | **Execute Permission Requested (EXE):** When set, indicates Execute permission was requested by the faulted read request.<br>This field is relevant only when the PP field and T field are both Set.<br>Hardware implementations not supporting PASID (PASID field Clear in Extended Capability register) implement this field as RsvdZ.<br><br>**Power Well:** prst |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 29 | 0x0<br>RO/V | **Privilege Mode Requested (PRIV):** When set, indicates Supervisor privilege was requested by the faulted request.<br>This field is relevant only when the PP field is Set.<br>Hardware implementations not supporting PASID (PASID field Clear in Extended Capability register) implement this field as RsvdZ.<br><br>**Power Well:** prst |
| 28:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RO/V | **Source Identifier (SID):** Requester-id associated with the fault condition.<br>This field is relevant only when the F field is set.<br><br>**Power Well:** prst |

## 44.4.35 Invalidate Address Register (IVA_REG_0_0_0_VTDBAR)—Offset 500h

Register to provide the DMA address whose corresponding IOTLB entry needs to be invalidated through the corresponding IOTLB Invalidate register. This register is a write-only register.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 500h

**Default:** 0000000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:12 | 0x0<br>RW | **Address (ADDR):** Software provides the DMA address that needs to be page-selectively invalidated.<br>To make a page-selective invalidation request to hardware, software must first write the appropriate fields in this register, and then issue the appropriate page-selective invalidate command through the IOTLB_REG.<br>Hardware ignores bits 63:N, where N is the maximum guest address width (MGAW) supported.<br>A value returned on a read of this field is undefined. |
| 11:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW | **Invalidation Hint (IH):** The field provides hint to hardware about preserving or flushing the non-leaf (page-directory) entries that may be cached in hardware:<br>• 0 = Software may have modified both leaf and non-leaf page-table entries corresponding to mappings specified in the ADDR and AM fields. On a page-selective invalidation request, hardware must flush both the cached leaf and non-leaf page-table entries corresponding to the mappings specified by ADDR and AM fields.<br>• 1 = Software has not modified any non-leaf page-table entries corresponding to mappings specified in the ADDR and AM fields. On a page-selective invalidation request, hardware may preserve the cached non-leaf page-table entries corresponding to mappings specified by ADDR and AM fields.<br>A value returned on a read of this field is undefined. |
| 5:0 | 0x0<br>RW | **Address Mask (AM):** The value in this field specifies the number of low order bits of the ADDR field that must be masked for the invalidation operation.<br>This field enables software to request invalidation of contiguous mappings for size-aligned regions.<br>For example:..Mask ADDR bits Pages..Value masked invalidated.. 0 None 1.. 1 12 2.. 2 13:12 4.. 3 14:12 8.. 4 15:12 16<br>When invalidating mappings for super-pages, software must specify the appropriate mask value. For example, when invalidating mapping for a 2MB page, software must specify an address mask value of at least 9...Hardware implementations report the maximum supported mask value through the Capability register. |

## 44.4.36 IOTLB Invalidate Register (IOTLB_REG_0_0_0_VTDBAR)—Offset 508h

Register to invalidate IOTLB. The act of writing the upper byte of the IOTLB_REG with IVT field Set causes the hardware to perform the IOTLB invalidation.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [DEFVTDBAR] + 508h

**Default:** 0200000000000000h

**DEFVTDBAR Type:** Memory Mapped I/O Register (Size: 64 bits)
**DEFVTDBAR Reference:** B_CR_DEFVTDBAR_0_0_0_MCHBAR + 6C80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x0<br>RW/V | **Invalidate IOTLB (IVT):** Software requests IOTLB invalidation by setting this field.<br>Software must also set the requested invalidation granularity by programming the IIRG field. Hardware clears the IVT field to indicate the invalidation request is complete.<br>Hardware also indicates the granularity at which the invalidation operation was performed through the IAIG field.<br>Software must not submit another invalidation request through this register while the IVT field is Set, nor update the associated Invalidate Address register.<br>Software must not submit IOTLB invalidation requests when there is a context-cache invalidation request pending at this remapping hardware unit.<br>Hardware implementations reporting write-buffer flushing requirement (RWBF=1 in Capability register) must implicitly perform a write buffer flushing before invalidating the IOTLB. |
| 62 | 0x0<br>RO | **RSVD0:** Reserved |
| 61:60 | 0x0<br>RW | **IOTLB Invalidation Request Granularity (IIRG):** When requesting hardware to invalidate the IOTLB (by setting the IVT field), software writes the requested invalidation granularity through this field.<br>The following are the encodings for the field:<br>• 00 = Reserved<br>• 01 = Global invalidation request<br>• 10 = Domain-selective invalidation request. The target domain-id must be specified in the DID field<br>• 11 = Page-selective invalidation request. The target address, mask and invalidation hint must be specified in the Invalidate Address register, and the domain-id must be provided in the DID field<br>Hardware implementations may process an invalidation request by performing invalidation at a coarser granularity than requested.<br>Hardware indicates completion of the invalidation request by clearing the IVT field.<br>At this time, the granularity at which actual invalidation was performed is reported through the IAIG field. |
| 59 | 0x0<br>RO | **RSVD1:** Reserved |
| 58:57 | 0x1<br>RO/V | **IOTLB Actual Invalidation Granularity (IAIG):** Hardware reports the granularity at which an invalidation request was processed through this field when reporting invalidation completion (by clearing the IVT field).<br>The following are the encodings for this field<br>• 00 = Reserved. This indicates hardware detected an incorrect invalidation request and ignored the request. Examples of incorrect invalidation requests include detecting an unsupported address mask value in Invalidate Address register for page-selective invalidation requests<br>• 01 = Global Invalidation performed. This could be in response to a global, domain-selective, or page-selective invalidation request<br>• 10 = Domain-selective invalidation performed using the domain-id specified by software in the DID field. This could be in response to a domain-selective or a page-selective invalidation request<br>• 11 = Domain-page-selective invalidation performed using the address, mask and hint specified by software in the Invalidate Address register and domain-id specified in DID field. This can be in response to a page-selective invalidation request. |
| 56:50 | 0x0<br>RO | **RSVD2:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 49 | 0x0<br>RW | **Drain Reads (DR):** This field is ignored by hardware if the DRD field is reported as clear in the Capability register.<br>When the DRD field is reported as Set in the Capability register, the following encodings are supported for this field:<br>• 0 = Hardware may complete the IOTLB invalidation without draining any translated DMA read requests<br>• 1 = Hardware must drain DMA read requests. |
| 48 | 0x0<br>RW | **Drain Writes (DW):** This field is ignored by hardware if the DWD field is reported as Clear in the Capability register.<br>When the DWD field is reported as Set in the Capability register, the following encodings are supported for this field:<br>• 0 = Hardware may complete the IOTLB invalidation without draining DMA write requests<br>• 1 = Hardware must drain relevant translated DMA write requests. |
| 47:32 | 0x0<br>RW | **Domain-ID (DID):** Indicates the ID of the domain whose IOTLB entries need to be selectively invalidated.<br>This field must be programmed by software for domain-selective and page-selective invalidation requests.<br>The Capability register reports the domain-id width supported by hardware.<br>Software must ensure that the value written to this field is within this limit. Hardware ignores and not implements bits 47:(32+N), where N is the supported domain-id width reported in the Capability register. |
| 31:0 | 0x0<br>RO | **RSVD3:** Reserved |

## 44.5 Sideband Registers

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space using the Sideband Base Address Register (BAR) DEFVTDBAR. It is located in Host Root Memory Space MCHBAR +offset 6C80h.

See Section 44.3.123, "Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR)—Offset 6C80h" on page 1316.

### 44.5.1 UCELOG—Offset 64h

Uncorrectable Error Log This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 64h
**Host Memory Space:** SBREG_BAR + 0x100000 + 64h

**Default:** 00000800h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RSV | **Rsvd_UCELOG_1:** Reserved |
| 21:20 | 0x0<br>RO/V/P | **UCERNK:** Rank Address: This is the rank address that contained the error |
| 19:16 | 0x0<br>RO/V/P | **UCEBNK:** Bank Address: This is the bank address that contained the error. DDR3: UCEBNK[3] will always be 0 UCEBNK[2:0] = BA[2:0] DDR4: UCEBNK[3:2] = BG[1:0] UCEBNK[1:0] = BA[1:0] |
| 15 | 0x0<br>RO/V/P | **UCEAP:** Auto Pre-charge was issued on the read that contained the error. This will always be 0 since AP is not supported. |
| 14 | 0x0<br>RO/V/P | **UCELAST:** Last Burst: If this bit is set, then the error occurred within that last four DQ bus bursts before a DQ bus turn-around cycle. |
| 13 | 0x0<br>RO/V/P | **UCEFIRST:** First Burst: If this bit is set, then the error occurred within the first four DQ bus bursts after a DQ bus turn-around cycle. |
| 12 | 0x0<br>RO/V/P | **UCEA5:** Address bit 5: This is A[5] from the original request |
| 11 | 0x1<br>RO/V/P | **UCEBL8:** Burst Length 8: 0 = Burst Chop 4, 1 = Burst Length 8. This will always be 1 for DNV. |
| 10:8 | 0x0<br>RO/V/P | **UCEBOFF:** Burst Offset: This is the burst number that contained the error. Note that all 8 bursts are always numbered starting from 0. Thus, even when the critical chunk causes the bursts to occur in the logical order of 4,5,6,7,0,1,2,3, UCEBOFF will still number these starting from 0. So, for example, if an error is detected on logical byte 0, it is possible that this may be recorded as byte offset 4 with critical chunk reordering. |
| 7:0 | 0x0<br>RO/V/P | **UCESYND:** ECC Syndrome: This is the 8-bit syndrome output of the ECC checker logic. For DNV, this field does not provide useful information and is not reliable. |

## 44.5.2 UCEADDR—Offset 68h

Uncorrectable Error Address This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 68h
**Host Memory Space:** SBREG_BAR + 0x100000 + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **UCEADDR:** Address Log: Captures the PMI address A[35:6] of the UCE error location. |

## 44.5.3 SBELOG—Offset 6Ch

Correctable (Single Bit) Error Log This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 6Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 6Ch

**Default:** 00000800h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RSV | **Rsvd_SBELOG_1:** Reserved |
| 21:20 | 0x0<br>RO/V/P | **SBERNK:** Rank Address: This is the rank address that contained the error |
| 19:16 | 0x0<br>RO/V/P | **SBEBNK:** Bank Address: This is the bank address that contained the error. DDR3: UCEBNK[3] will always be 0 UCEBNK[2:0] = BA[2:0] DDR4: UCEBNK[3:2] = BG[1:0] UCEBNK[1:0] = BA[1:0] |
| 15 | 0x0<br>RO/V/P | **Reserved:** Reserved |
| 14 | 0x0<br>RO/V/P | **SBELAST:** Last Burst: If this bit is set, then the error occurred within that last four DQ bus bursts before a DQ bus turn-around cycle. |
| 13 | 0x0<br>RO/V/P | **SBEFIRST:** First Burst: If this bit is set, then the error occurred within the first four DQ bus bursts after a DQ bus turn-around cycle. |
| 12 | 0x0<br>RO/V/P | **SBEA5:** Address bit 5: This is A[5] from the original request. In 32 bit mode this field is undefined. |
| 11 | 0x1<br>RO/V/P | **Reserved:** Reserved |
| 10:8 | 0x0<br>RO/V/P | **SBEBOFF:** Burst Offset: This is the burst number that contained the error. Note that all 8 bursts are always numbered starting from 0. |
| 7:0 | 0x0<br>RO/V/P | **SBESYND:** ECC Syndrome: This is the 8-bit syndrome output of the ECC checker logic |

### 44.5.4 SBEADDR—Offset 70h

Correctable Error Address This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 70h
**Host Memory Space:** SBREG_BAR + 0x100000 + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **SBEADDR:** Address Log: Captures the PMI address A[35:6] of the SBE error location. |

## 44.5.5 SBECNT0—Offset 74h

SBE Count for Rank 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 74h
**Host Memory Space:** SBREG_BAR + 0x100000 + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT0_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 0. Any write to this register will clear the counter to 0. |

## 44.5.6 SBECNT1—Offset 78h

SBE Count for Rank 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 78h
**Host Memory Space:** SBREG_BAR + 0x100000 + 78h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT1_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 1. Any write to this register will clear the counter to 0. |

## 44.5.7 SBECNT2—Offset 7Ch

SBE Count for Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 7Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 7Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT2_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 2. Any write to this register will clear the counter to 0. |

## 44.5.8 SBECNT3—Offset 80h

SBE Count for Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 80h
**Host Memory Space:** SBREG_BAR + 0x100000 + 80h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT3_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 3. Any write to this register will clear the counter to 0. |

## 44.5.9 SBEACC0—Offset 84h

SBE Accumulator for Rank0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 84h
**Host Memory Space:** SBREG_BAR + 0x100000 + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC0_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.10 SBEACC1—Offset 88h

SBE Accumulator for Rank1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 88h
**Host Memory Space:** SBREG_BAR + 0x100000 + 88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC1_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.11 SBEACC2—Offset 8Ch

SBE Accumulator for Rank2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 8Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 8Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC2_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.12 SBEACC3—Offset 90h

SBE Accumulator for Rank3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 90h
**Host Memory Space:** SBREG_BAR + 0x100000 + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC3_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.13 DPATROL_SCRUB_CFG—Offset B0h

Patrol Scrub Config Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + B0h
**Host Memory Space:** SBREG_BAR + 0x100000 + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **PAT_SCRUB_ENABLE:** Bit to enable/disable patrol scrub<br>• 0: Disable patrol scrub<br>• 1: Enable patrol scrub |
| 30 | 0x0<br>RW/P | **DIMM1_COL_ADDR_BITS:** Total number of column bits for devices on rank 0 = 10 + DIMM1_COL_ADDR_BITS<br>• 0: DDR4 and DDR3<br>• 1: Only DDR3 |
| 29 | 0x0<br>RW/P | **DIMM0_COL_ADDR_BITS:** Total number of column bits for devices on rank 0 = 10 + DIMM0_COL_ADDR_BITS<br>• 0: DDR4 and DDR3<br>• 1: Only DDR3 |
| 28:23 | 0x0<br>RW/P | **DIMM1_MAX_ROW_ADDR:** Maximum row address for DIMM1 - bits 17 to 12. Bits 11:0 are assumed to all be 1. |
| 22:17 | 0x0<br>RW/P | **DIMM0_MAX_ROW_ADDR:** Maximum row address for DIMM0 - bits 17 to 12. Bits 11:0 are assumed to all be 1. |
| 16 | 0x0<br>RW/P | **PATSCRUB_FORCE_WRITEBACK:** When set to 1, force write back of data with no errors. When set to 0, don't write back data that had no errors. |
| 15:0 | 0x0<br>RW/P | **MAX_PATSCRUB_DEBIT_CNT:** Maximum number of debit patrol scrub operations. Once the patrol scrub debit counter exceeds this value, patrol scrubs will be issued until the debit counter reaches 0. Set 0 to disable debit mode. |

## 44.5.14 UCELOG1—Offset CCh

This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + CCh
**Host Memory Space:** SBREG_BAR + 0x100000 + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RSV | **RSVD (reg_RSVD):** Reserved |
| 26:18 | 0x0<br>RO/V/P | **UCE_COL:** Column address bits above A[5] |
| 17:0 | 0x0<br>RO/V/P | **UCE_ROW:** Row bits |

## 44.5.15 SBELOG1—Offset D0h

Single bit error log 1 This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + D0h
**Host Memory Space:** SBREG_BAR + 0x100000 + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RSV | **RSVD (reg_RSVD):** Reserved |
| 26:18 | 0x0<br>RO/V/P | **SBE_COL:** Column bits [11,9:2], in 32 bit mode SBE_COL[0] will always be 0, SBE_COL[1] contains information for SBEBOFF[2] |
| 17:0 | 0x0<br>RO/V/P | **SBE_ROW:** Row bits |

## 44.5.16 CORRERRTHRSHLD0—Offset 100h

Correctable Error Threshold for Rank 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 100h
**Host Memory Space:** SBREG_BAR + 0x100000 + 100h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD0_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 0. When the error accumulation in SBECNT0 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.17 CORRERRTHRSHLD1—Offset 104h

Correctable Error Threshold for Rank 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 104h
**Host Memory Space:** SBREG_BAR + 0x100000 + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD1_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 1. When the error accumulation in SBECNT1 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.18 CORRERRTHRSHLD2—Offset 108h

Correctable Error Threshold for Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 108h
**Host Memory Space:** SBREG_BAR + 0x100000 + 108h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD2_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 2. When the error accumulation in SBECNT2 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.19 CORRERRTHRSHLD3—Offset 10Ch

Correctable Error Threshold for Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 10Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 10Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD3_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 3. When the error accumulation in SBECNT3 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.20 LEAKY_BUCKET_CFG0—Offset 110h

Leaky Bucket Configuration Rank 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 110h
**Host Memory Space:** SBREG_BAR + 0x100000 + 110h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 0 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 0, CNTR_HI. |

## 44.5.21 LEAKY_BUCKET_CFG1—Offset 114h

Leaky Bucket Configuration Rank 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 114h
**Host Memory Space:** SBREG_BAR + 0x100000 + 114h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 1 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 1, CNTR_HI. |

## 44.5.22 LEAKY_BUCKET_CFG2—Offset 118h

Leaky Bucket Configuration Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 118h
**Host Memory Space:** SBREG_BAR + 0x100000 + 118h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 2 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 2, CNTR_HI. |

## 44.5.23 LEAKY_BUCKET_CFG3—Offset 11Ch

Leaky Bucket Configuration Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 11Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 11Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 3 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 3, CNTR_HI. |

## 44.5.24 FERRNERR—Offset 120h

First Error and Next Error Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 120h
**Host Memory Space:** SBREG_BAR + 0x100000 + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RSV | **Rsvd_FERRNERR_1 (Rsvd_FERRNERR_3):** Reserved |
| 20 | 0x0<br>RO/V/P | **FERRUCEPAT:** First Error Uncorrectable Patrol Scrub Read Error Detected: When set, the first error detected was an uncorrectable (multiple bit) error on the patrol scrub read from memory. |
| 19 | 0x0<br>RO/V/P | **FERRAPE:** First Error Uncorrectable DDR4 Command Address Parity Error: When set, the first error detected was the assertion of the xxDDR3_0_ALERT_N pin indicating that one or more DRAMs detected a Command/Address parity error. |
| 18 | 0x0<br>RO/V/P | **FERRWPE:** First Error Uncorrectable Write Parity Error Detected: When set, the first error detected was an uncorrectable (parity) error on the write data received from the Bunit. |
| 17 | 0x0<br>RO/V/P | **FERRUCE:** First Error Uncorrectable Read Error Detected: When set, the first error detected was an uncorrectable (multiple bit) error on the read data from memory. |
| 16 | 0x0<br>RO/V/P | **FERRSBE:** First Error Correctable Read Error Detected: When set, the first error detected was a single bit error on the read data from memory and the error was corrected. |
| 15:5 | 0x0<br>RSV | **Rsvd_FERRNERR_0 (Rsvd_FERRNERR_1):** Reserved |
| 4 | 0x0<br>RO/V/P | **NERRUCEPAT:** Next Error Uncorrectable Patrol Scrub Read Error: When set, the second error detected was an uncorrectable (mutliple bit) error on the patrol scrub read data from memory. |
| 3 | 0x0<br>RO/V/P | **NERRAPE:** Next Error Uncorrectable DDR4 Command Address Parity Error: When set, the second error detected was the assertion of the xxDDR3_0_ALERT_N pin indicating that one or more DRAMs detected a Command/Address parity error. Note that this bit can never be set as the same time as FERRAPE |
| 2 | 0x0<br>RO/V/P | **NERRWPE:** Next Error Uncorrectable Write Parity Error Detected: When set, the second error detected was an uncorrectable (parity) error on the write data received from the Bunit. |
| 1 | 0x0<br>RO/V/P | **NERRUCE:** Next Error Uncorrectable Read Error Detected: When set, the second error detected was an uncorrectable (multiple bit) error on the read data from memory. |
| 0 | 0x0<br>RO/V/P | **NERRSBE:** Next Error Correctable Read Error Detected: When set, the second error detected was a single bit error on the read data from memory and the error was corrected. |

## 44.5.25 DERRSTS—Offset 124h

Error Status Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 124h
**Host Memory Space:** SBREG_BAR + 0x100000 + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RSV | **Rsvd_DERRSTS_0 (Rsvd_DERRSTS_1):** Reserved |
| 6 | 0x0<br>RW/1C/V/P | **CAP_ERR_STATE:** Channel Command Address Parity Error State. 1 = CAP error seen, error state entered. All PMI reads will be returned with bad parity. 0 = No CAP error seen. Normal PMI operation. |
| 5 | 0x0<br>RO/V/P | **ALERT:** Current state of the channel ALERT_N pin, captured here as active high: 1 = Error being driven by at least one DRAM device 0 = No error currently detected by any DRAM device |
| 4 | 0x0<br>RW/1C/V/P | **UCEPAT:** Uncorrectable (multiple bit) patrol scrub error: When set, an uncorrectable (multiple bit error was detected on the read data from memory. |
| 3 | 0x0<br>RW/1C/V/P | **APE:** Uncorrectable DDR4 Command/Address Parity Error: When set, an uncorrectable DDR4 command/address parity error was detected by one or more of the DRAMs on channel 0. |
| 2 | 0x0<br>RW/1C/V/P | **WPE:** Uncorrectable (Parity) Write Error: When set, an uncorrectable (parity) error was detected on the write data received from the Bunit. |
| 1 | 0x0<br>RW/1C/V/P | **UCE:** Uncorrectable (Double Bit) Read Error: When set, a double bit error was detected on the read data from memory. |
| 0 | 0x0<br>RW/1C/V/P | **SBE:** Correctable (Single Bit) Read Error: When set, a single bit error was detected on the read data from memory and the error was corrected. |

## 44.5.26 DERRMSKSEV—Offset 128h

Error Mask and Severity

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 128h
**Host Memory Space:** SBREG_BAR + 0x100000 + 128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_2:** Reserved |
| 20 | 0x0<br>RW/P | **MSK_UCEPAT:** Uncorrectable patrol scrub error mask: When set, uncorrectable patrol scrub ECC data errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 19 | 0x0<br>RW/P | **MSK_APE:** Uncorrectable command/address parity error mask: When set, uncorrectable DDR4 command/address parity errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 18 | 0x0<br>RW/P | **MSK_WDBR:** Uncorrectable (parity) write data buffer read error mask: When set, uncorrectable (pairty) write data buffer read errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 17 | 0x0<br>RW/P | **MSK_UCE:** Uncorrectable (multiple bit) error mask: When set, uncorrectable (multiple bit) read and patrol scrub errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 16 | 0x0<br>RW/P | **MSK_SBE:** Correctable (single bit) error mask: When set, correctable (single bit) read and patrol scrub errors will not be logged in DERRESTS[0] and DO_SERR_SEV messages will not be sent to the IEH. |
| 15:5 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_1:** Reserved |
| 4 | 0x0<br>RW/P | **SEV_UCEPAT:** Uncorrectable (Multiple Bit) Patrol Scrub Read Error Severity: When set, UCE errors logged in DERRSTS[5] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 3 | 0x0<br>RW/P | **SEV_APE:** Uncorrectable DDR4 Command/Address Parity Error Severity: When set, APE errors logged in DERRSTS[3] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 2 | 0x0<br>RW/P | **SEV_WDBR:** Uncorrectable (Parity) Write Error Severity: When set, WPE errors logged in DERRSTS[2] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 1 | 0x0<br>RW/P | **SEV_UCE:** Uncorrectable (Multiple Bit) Read Error Severity: When set, UCE errors logged in DERRSTS[1] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 0 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_0:** Reserved |

## 44.5.27 DERRCNTSEL—Offset 12Ch

Uncorrectable Error Count Select

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 12Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 12Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RSV | **Rsvd_DERRCNTSEL_2:** Reserved |
| 4 | 0x0<br>RW/P | **SEL_UCEPAT:** Uncorrectable (Multiple Bit) Patrol Scrub Read Error Count Select: When set, UCE errors logged in DERRSTS[1] will be counted in DERRCNT.UNCCNT1. |
| 3 | 0x0<br>RW/P | **SEL_APE:** Uncorrectable DDR4 Command/Address Parity Error Count Select: When set, APE0 errors logged in DERRSTS[3] will be counted in DERRCNT.UNCCNT0. |
| 2 | 0x0<br>RW/P | **SEL_WPE:** Uncorrectable (Parity) Write Error Count Select: When set, WPE0 errors logged in DERRSTS[2] will be counted in DERRCNT.UNCCNT0. |
| 1 | 0x0<br>RW/P | **SEL_UCE:** Uncorrectable (Multiple Bit) Read Error Count Select: When set, UCE0 errors logged in DERRSTS[1] will be counted in DERRCNT.UNCCNT0. |
| 0 | 0x0<br>RSV | **Rsvd_DERRCNTSEL_0:** Reserved |

## 44.5.28 DERRCNT—Offset 130h

Uncorrectable Error Count

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 130h
**Host Memory Space:** SBREG_BAR + 0x100000 + 130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RSV | **Rsvd_DERRCNT:** Reserved |
| 7:0 | 0x0<br>RW/1C/V/P | **UNCCNT:** Uncorrectable Error Count: The count of uncorrectable errors that have occurred since the last time the count was cleared. The count limit is 128 (80h) and the count is cleared by writing ones to all bits (FFh). The type of uncorrectable errors that are counted is determined by the DERRCNTSEL register. Additionally, if two uncorrectable errors occur in the same clock cycle, only one type will be counted, even if multiple types are selected. |

## 44.5.29 ERRINJCTL—Offset 134h

Error Injection Control

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 134h
**Host Memory Space:** SBREG_BAR + 0x100000 + 134h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/P | **INJECCPAR:** Error Injection ECC/Parity[7:0]: Error injection ECC/parity. Zero bits indicate no injection; one bits indicate that the corresponding read/write ECC or parity bit should be inverted. |
| 23:20 | 0x0<br>RSV | **Rsvd_ERRINJCTL_2:** Reserved |
| 19 | 0x0<br>RW/P | **INJAMSK4:** Error Injection Address Mask[4]: This is the mask bit for error injection address bit 4; address mask bits 36:5 are located in the ERRINJAMSK register. |
| 18 | 0x0<br>RW/P | **INJADDR4:** Error Injection Address[4]: This is error injection address bit 4; address bits 36:5 are located in the ERRINJADDR register. |
| 17 | 0x0<br>RW/P | **INJAMSK3:** Error Injection Address Mask[3]: This is the mask bit for error injection address bit 3; address mask bits 36:5 are located in the ERRINJAMSK register. |
| 16 | 0x0<br>RW/P | **INJADDR3:** Error Injection Address[3]: This is error injection address bit 3; address bits 35:4 are located in the ERRINJADDR register. |
| 15:14 | 0x0<br>RW/P | **INJTRIGSEL:** Error Injection Trigger Selection: 0x = Disable error injection trigger sources 10 = Enable error injection when trigger source 0 (ddr_fsta_trigev[0]) is asserted 11 = Enable error injection when trigger source 1 (ddr_fsta_trigev[1]) is asserted. When setting this field to 1X, the INJEN field should be set to 0 since it will be set automatically when the trigger asserts. |
| 13:12 | 0x0<br>RW/P | **INJCAP:** Command address parity injection per dclk half |
| 11:8 | 0x0<br>RW/P | **INJSRC:** Error Injection Source[3:0]: The selected requestor source ID match against when selecting the transaction for error injection. |
| 7 | 0x0<br>RW/P | **INJPAT:** Inject on patrol scrub request. Address match/mask logic is not used here. Error injection will occur on the next patrol scrub once INJEN is set. |
| 6 | 0x0<br>RW/P | **INJREAD:** Error Injection in the Read Data Path: 0 = Disable 1 = Enabled |
| 5 | 0x0<br>RW/P | **INJWRITE:** Error Injection in the Write Data Path: 0 = Disable 1 = Enabled |
| 4 | 0x0<br>RW/P | **INJECC:** Error Injection in ECC or Parity Path Selection: 0 = Inject in the Data/Parity Path 1 = Inject in the Data/ECC Path |
| 3 | 0x0<br>RW/P | **INJSRCEN:** Error Injection Source Enable: When set, the error injection will be performed only for the specified requestor source. |
| 2 | 0x0<br>RW/P | **INJADREN:** Error Injection Address Enable: When set, the error injection will be performed only for addresses that match the address specified in the ERRINJADDR and ERRINJAMSK registers. |
| 1 | 0x0<br>RW/P | **INJONCE:** Error Injection Once: When set, a single error will be injected after error injection is enabled. Otherwise, error injection will be continuous until disabled. Note that error injection has a minimum temporal granularity of one dunit clock cycle. Thus, if the address match/match bits allow injections on multiple bursts, it is possible to inject up to 4 errors (across the 4 bursts per dunit cycle) even with this bit set. To ensure predictable single error injection INJAMSK3, INJAMSK4 and ERRINJAMSK.INJAMSK[0] must be set to 0. |
| 0 | 0x0<br>RW/V/P | **INJEN:** Error Injection Enable: 0 - Error injection is disabled 1 - Error injection is enabled. This bit is self clearing when inject once is selected and the single error injection has been performed. This bit should be set to 0 when INJTRIGSEL is set to 1X. Upon assertion, the trigger will automatically set the INJEN field. |

## 44.5.30 ERRINJADDR—Offset 138h

Error Injection Address

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 138h
**Host Memory Space:** SBREG_BAR + 0x100000 + 138h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJADDR:** Error Injection PMI Address[36:5] |

## 44.5.31 ERRINJAMSK—Offset 13Ch

Error Injection Address Mask

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 13Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 13Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJAMSK:** Error Injection PMI Address Mask[36:5]: For each bit that is 0 in the address mask, the corresponding PMI address bit will be compared against the same bit in ERRINJADDR. |

## 44.5.32 ERRINJDATA0—Offset 140h

Error Injection Data 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 140h
**Host Memory Space:** SBREG_BAR + 0x100000 + 140h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJDATA0:** Error Injection Data[31:0]: Zero bits indicate no injection; one bits indicate that the corresponding read/write data bit should be inverted. |

## 44.5.33 ERRINJDATA1—Offset 144h

Error Injection Data 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 144h
**Host Memory Space:** SBREG_BAR + 0x100000 + 144h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJDATA1:** Error Injection Data[63:32]: Zero bits indicate no injection; one bits indicate that the corresponding read/write data bit should be inverted. |

## 44.5.34 DCS—Offset 170h

DRAM Control and Status

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 170h
**Host Memory Space:** SBREG_BAR + 0x100000 + 170h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RSV | **Rsvd_DCS_5:** Reserved |
| 28 | 0x0<br>RW | **Reserved:** Reserved |
| 27:24 | 0x0<br>RSV | **Rsvd_DCS_4:** Reserved |
| 23 | 0x0<br>RW/V/P | **SCRAMBLE_ENABLE:** Data Scrambler Enable:<br>When set to 1 the data lanes (DDR3_DQ[63:0]) and the ECC lanes (DDR3_DQECC[7:0]) will be scrambled, otherwise no scrambling will occur. |
| 22 | 0x0<br>RSV | **Rsvd_DCS_3:** Reserved |
| 21 | 0x0<br>RW/P | **MPRMODE:** Mask Chip-Selects for all commands except read commands and MRS commands |
| 20 | 0x0<br>RW/P | **Reserved:** Reserved |
| 19:18 | 0x0<br>RSV | **Rsvd_DCS_2:** Reserved |
| 17:16 | 0x0<br>RW/P | **Reserved:** Reserved |
| 15:13 | 0x0<br>RSV | **Rsvd_DCS_1:** Reserved |
| 12 | 0x0<br>RW/P | **ODTMODE:** ODT Control Mode:<br>• 0 - Dunit auto controls the ODT pins based on DRAM Write transactions.<br>• 1 - The value of ODTVAL above directly controls the ODT pins. |
| 11:8 | 0x0<br>RW/P | **ODTVAL:** ODT Control Value: Each bit of ODTVAL directly controls an ODT pin, when ODTMODE is set to 1. Used only during init flow by BIOS. |
| 7:5 | 0x0<br>RSV | **Rsvd_DCS_0:** Reserved |
| 4 | 0x0<br>RW/P | **CKEMODE:** CKE Control Mode:<br>• 0 - Dunit auto controls the CKE pins based on Power-Down and Self Refresh entry and exit.<br>• 1 - The value of CKEVAL above directly controls the CKE pins. |
| 3:0 | 0x0<br>RW/P | **CKEVAL:** CKE Control Value: Each bit of CKEVAL directly controls a CKE pin, when CKEMODE is set to 1. Used only during init flow by BIOS. |

## 44.5.35 DECCCTRL—Offset 180h

Dunit ECC Control

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 180h
**Host Memory Space:** SBREG_BAR + 0x100000 + 180h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:26 | 0x0<br>RSV | **Rsvd_DECCCTRL_1:** Reserved |
| 25 | 0x0<br>RW/P | **RD2CLRACC:** Read to Clear C[1:0]SBEACC Registers:<br>• 0 = C[1:0]SBEACC Registers are cleared when written<br>• 1 = C[1:0]SBEACC Registers are cleared when written or read |
| 24:18 | 0x0<br>RSV | **Rsvd_DECCCTRL_0:** Reserved |
| 17 | 0x0<br>RW/P | **ENCBGEN:** Enable Generation of ECC Check Bits:<br>• 0 - Disable generation of ECC Check Bits<br>ECC for Bunit is 0<br>ECC for CPGC is generated by the CPCC<br>• 1 - Enable generation of ECC Check Bits<br>ECC is generated for both the Bunit and the CPGC |
| 16 | 0x0<br>RW/P | **ECCOVREN:** ECC Check Bits Override Enable:<br>• 0 - Normal ECC is generated.<br>• 1 - ECC override enabled, the value in ECCOVR will be used for the ECC code. |
| 15:8 | 0x0<br>RW/P | **ECCOVR:** ECC Check Bits Override Value:<br>The ECC override code to be used on writes if enabled by ECCOVREN. |
| 7:5 | 0x0<br>RW/P | **UCE_FILTER:** Uncorrectable Error Filter:<br>• 000: Normal mode; All 8 byte ECC checks/corrections are performed independently, no anti-aliasing filters are applied.<br>• 011: An uncorrectable error that is detected in any 8 byte ECC check will cause all four parallel 8 byte ECC checks to be treated as uncorrectable errors. |
| 4 | 0x0<br>RW/P | **DPAREN:** Data Parity Enable:<br>When set, data path parity is enabled. The write data parity is checked at the point when ECC is generated and read data parity is generated at the point that ECC is checked. |
| 3 | 0x0<br>RW/P | **CAPERREN:** Enable Command Address Parity Error Detection:<br>When set, command address parity errors are detected and signaled to the Bunit. |
| 2 | 0x0<br>RW/P | **DERREN:** Enable Uncorrectable Error Notification:<br>When set, uncorrectable ECC errors are signaled to the Bunit and patrol scrub. |
| 1 | 0x0<br>RW/P | **SERREN:** Enable Correctable Error Notification:<br>When set, correctable ECC errors are signaled to the Bunit and patrol scrub which indicates that the read data needs to be written back to memory. This causes the DRAM cell with the single bit error to be scrubbed. |
| 0 | 0x0<br>RW/V/P | **ECCEN:** ECC Enable: When set, single bit error correction and double bit error detection (SEC/DED ECC) is enabled. The ECC is an 8 bit protection code for every 64 bits of data stored in DRAM. ECC is generated on writes and checked on reads, requiring a 72 bit DDR3 data bus width. |

## 44.5.36 LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch

Leaky Bucket Counter Upper

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 18Ch
**Host Memory Space:** SBREG_BAR + 0x100000 + 18Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/P | **CNTR_UPPER_RANK3:** Rank 3 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK3 + 1)* (LEAKY_BUCKET_CFG3.CNTR_HI + 1) * (LEAKY_BUCKET_CFG3.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 23:16 | 0x0<br>RW/P | **CNTR_UPPER_RANK2:** Rank 2 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK2 + 1)* (LEAKY_BUCKET_CFG2.CNTR_HI + 1) * (LEAKY_BUCKET_CFG2.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 15:8 | 0x0<br>RW/P | **CNTR_UPPER_RANK1:** Rank 1 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK1 + 1)* (LEAKY_BUCKET_CFG1.CNTR_HI + 1) * (LEAKY_BUCKET_CFG1.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 7:0 | 0x0<br>RW/P | **CNTR_UPPER_RANK0:** Rank 0 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK0 + 1)* (LEAKY_BUCKET_CFG0.CNTR_HI + 1) * (LEAKY_BUCKET_CFG0.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |

## 44.5.37 DPATROL_SCRUB_TMR—Offset 194h

Patrol Scrub Timer Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x10] + 194h
**Host Memory Space:** SBREG_BAR + 0x100000 + 194h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RW/P | **PATSCRTIMER:** Patrol scrub interval timer in DCLKs (DRAM cmd clk divided by 2). A new patrol scrub is scheduled every time the timer ticks to the value in this register. Any non zero value is valid |

## 44.5.38 UCELOG—Offset 64h

Uncorrectable Error Log This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 64h
**Host Memory Space:** SBREG_BAR + 0x120000 + 64h

**Default:** 00000800h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RSV | **Rsvd_UCELOG_1:** Reserved |
| 21:20 | 0x0<br>RO/V/P | **UCERNK:** Rank Address: This is the rank address that contained the error |
| 19:16 | 0x0<br>RO/V/P | **UCEBNK:** Bank Address: This is the bank address that contained the error. DDR3: UCEBNK[3] will always be 0 UCEBNK[2:0] = BA[2:0] DDR4: UCEBNK[3:2] = BG[1:0] UCEBNK[1:0] = BA[1:0] |
| 15 | 0x0<br>RO/V/P | **Reserved:** Reserved |
| 14 | 0x0<br>RO/V/P | **UCELAST:** Last Burst: If this bit is set, then the error occurred within that last four DQ bus bursts before a DQ bus turn-around cycle. |
| 13 | 0x0<br>RO/V/P | **UCEFIRST:** First Burst: If this bit is set, then the error occurred within the first four DQ bus bursts after a DQ bus turn-around cycle. |
| 12 | 0x0<br>RO/V/P | **UCEA5:** Address bit 5: This is A[5] from the original request |
| 11 | 0x1<br>RO/V/P | **Reserved:** Reserved |
| 10:8 | 0x0<br>RO/V/P | **UCEBOFF:** Burst Offset: This is the burst number that contained the error. Note that all 8 bursts are always numbered starting from 0. |
| 7:0 | 0x0<br>RO/V/P | **UCESYND:** ECC Syndrome: This is the 8-bit syndrome output of the ECC checker logic. For DNV, this field does not provide useful information and is not reliable. |

## 44.5.39 UCEADDR—Offset 68h

Uncorrectable Error Address This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 68h
**Host Memory Space:** SBREG_BAR + 0x120000 + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **UCEADDR:** Address Log: Captures the PMI address A[35:6] of the UCE error location. |

## 44.5.40 SBELOG—Offset 6Ch

Correctable (Single Bit) Error Log This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 6Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 6Ch

**Default:** 00000800h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RSV | **Rsvd_SBELOG_1:** Reserved |
| 21:20 | 0x0<br>RO/V/P | **SBERNK:** Rank Address: This is the rank address that contained the error |
| 19:16 | 0x0<br>RO/V/P | **SBEBNK:** Bank Address: This is the bank address that contained the error. DDR3: UCEBNK[3] will always be 0 UCEBNK[2:0] = BA[2:0] DDR4: UCEBNK[3:2] = BG[1:0] UCEBNK[1:0] = BA[1:0] |
| 15 | 0x0<br>RO/V/P | **Reserved:** Reserved |
| 14 | 0x0<br>RO/V/P | **SBELAST:** Last Burst: If this bit is set, then the error occurred within that last four DQ bus bursts before a DQ bus turn-around cycle. |
| 13 | 0x0<br>RO/V/P | **SBEFIRST:** First Burst: If this bit is set, then the error occurred within the first four DQ bus bursts after a DQ bus turn-around cycle. |
| 12 | 0x0<br>RO/V/P | **SBEA5:** Address bit 5: This is A[5] from the original request. In 32 bit mode this field is undefined. |
| 11 | 0x1<br>RO/V/P | **Reserved:** Reserved |
| 10:8 | 0x0<br>RO/V/P | **SBEBOFF:** Burst Offset: This is the burst number that contained the error. Note that all 8 bursts are always numbered starting from 0. |
| 7:0 | 0x0<br>RO/V/P | **SBESYND:** ECC Syndrome: This is the 8-bit syndrome output of the ECC checker logic |

## 44.5.41 SBEADDR—Offset 70h

Correctable Error Address This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 70h
**Host Memory Space:** SBREG_BAR + 0x120000 + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **SBEADDR:** Address Log: Captures the PMI address A[35:6] of the SBE error location. |

## 44.5.42 SBECNT0—Offset 74h

SBE Count for Rank 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 74h
**Host Memory Space:** SBREG_BAR + 0x120000 + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT0_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 0. Any write to this register will clear the counter to 0. |

## 44.5.43 SBECNT1—Offset 78h

SBE Count for Rank 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 78h
**Host Memory Space:** SBREG_BAR + 0x120000 + 78h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT1_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 1. Any write to this register will clear the counter to 0. |

## 44.5.44 SBECNT2—Offset 7Ch

SBE Count for Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 7Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 7Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT2_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 2. Any write to this register will clear the counter to 0. |

## 44.5.45 SBECNT3—Offset 80h

SBE Count for Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 80h
**Host Memory Space:** SBREG_BAR + 0x120000 + 80h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RSV | **Rsvd_SBECNT3_0:** Reserved |
| 16 | 0x0<br>RW/C/V/P | **CORRERRORSTATUS:** The per rank status bits are set when a rank's error count equals the threshold. The status bits are what tell the BIOS/BMC what caused SMI/ERR[0] to get triggered. Once the status bits are set, they will remain set until cleared by BIOS and will cause the SMI/ERR[0] to get signaled to the cores. |
| 15:0 | 0x0<br>RW/C/V/P | **CNT:** Single Bit Error Count for Rank 3. Any write to this register will clear the counter to 0. |

## 44.5.46 SBEACC0—Offset 84h

SBE Accumulator for Rank0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 84h
**Host Memory Space:** SBREG_BAR + 0x120000 + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC0_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.47 SBEACC1—Offset 88h

SBE Accumulator for Rank1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 88h
**Host Memory Space:** SBREG_BAR + 0x120000 + 88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC1_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane: This allows for identification of the failing x4 or x8 device. This value is cleared by any write to this register. |

## 44.5.48 SBEACC2—Offset 8Ch

SBE Accumulator for Rank2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 8Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 8Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC2_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane:<br>This allows for identification of the failing x4 or x8 device. |

## 44.5.49 SBEACC3—Offset 90h

SBE Accumulator for Rank3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 90h
**Host Memory Space:** SBREG_BAR + 0x120000 + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RSV | **Rsvd_SBEACC3_0:** Reserved |
| 17:0 | 0x0<br>RW/C/V/P | **LANE:** Single Bit Accumulator per Nibble Lane:<br>This allows for identification of the failing x4 or x8 device. |

## 44.5.50 DPATROL_SCRUB_CFG—Offset B0h

Patrol Scrub Config Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + B0h
**Host Memory Space:** SBREG_BAR + 0x120000 + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **PAT_SCRUB_ENABLE:** Bit to enable/disable patrol scrub<br>0: Disable patrol scrub<br>1: Enable patrol scrub |
| 30 | 0x0<br>RW/P | **DIMM1_COL_ADDR_BITS:** Total number of column bits for devices on rank 0 = 10 + DIMM1_COL_ADDR_BITS<br>0: DDR4 and DDR3<br>1: Only DDR3 |
| 29 | 0x0<br>RW/P | **DIMM0_COL_ADDR_BITS:** Total number of column bits for devices on rank 0 = 10 + DIMM0_COL_ADDR_BITS<br>0: DDR4 and DDR3<br>1: Only DDR3 |
| 28:23 | 0x0<br>RW/P | **DIMM1_MAX_ROW_ADDR:** Maximum row address for DIMM1 - bits 17 to 12. Bits 11:0 are assumed to all be 1. |
| 22:17 | 0x0<br>RW/P | **DIMM0_MAX_ROW_ADDR:** Maximum row address for DIMM0 - bits 17 to 12. Bits 11:0 are assumed to all be 1. |
| 16 | 0x0<br>RW/P | **PATSCRUB_FORCE_WRITEBACK:** When set to 1, force write back of data with no errors. When set to 0, don't write back data that had no errors. |
| 15:0 | 0x0<br>RW/P | **MAX_PATSCRUB_DEBIT_CNT:** Maximum number of debit patrol scrub operations. Once the patrol scrub debit counter exceeds this value, patrol scrubs will be issued until the debit counter reaches 0. Set 0 to disable debit mode. |

## 44.5.51 UCELOG1—Offset CCh

This register captures the log information when the corresponding channel DERRSTS.UCE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.UCE[1:0] is cleared to 0.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + CCh
**Host Memory Space:** SBREG_BAR + 0x120000 + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RSV | **RSVD (reg_RSVD):** Reserved |
| 26:18 | 0x0<br>RO/V/P | **UCE_COL:** Column address bits above A[5] |
| 17:0 | 0x0<br>RO/V/P | **UCE_ROW:** Row bits |

## 44.5.52 SBELOG1—Offset D0h

Single bit error log 1 This register captures the log information when the corresponding channel DERRSTS.SBE[1:0] is set to 1. This register is cleared and capturing of new errors is re-enabled when the corresponding channel DERRSTS.SBE[1:0] is cleared.

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + D0h
**Host Memory Space:** SBREG_BAR + 0x120000 + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RSV | **RSVD (reg_RSVD):** Reserved |
| 26:18 | 0x0<br>RO/V/P | **SBE_COL:** Column bits [11,9:2], in 32 bit mode SBE_COL[0] will always be 0, SBE_COL[1] contains information for SBEBOFF[2] |
| 17:0 | 0x0<br>RO/V/P | **SBE_ROW:** Row bits |

## 44.5.53 CORRERRTHRSHLD0—Offset 100h

Correctable Error Threshold for Rank 0

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 100h
**Host Memory Space:** SBREG_BAR + 0x120000 + 100h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD0_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 0. When the error accumulation in SBECNT0 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.54 CORRERRTHRSHLD1—Offset 104h

Correctable Error Threshold for Rank 1

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 104h
**Host Memory Space:** SBREG_BAR + 0x120000 + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD1_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 1. When the error accumulation in SBECNT1 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.55 CORRERRTHRSHLD2—Offset 108h

Correctable Error Threshold for Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 108h
**Host Memory Space:** SBREG_BAR + 0x120000 + 108h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD2_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 2. When the error accumulation in SBECNT2 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.56 CORRERRTHRSHLD3—Offset 10Ch

Correctable Error Threshold for Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 10Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 10Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **Rsvd_CORRERRTHRSHLD3_0:** Reserved |
| 15:0 | 0x0<br>RW/P | **THRSH_RNK:** Correctable error threshold for rank 3. When the error accumulation in SBECNT3 hits this value, then the CORRERRSTATUS is set and a DO_SMI message is sent if the threshold value is greater than 0. |

## 44.5.57 LEAKY_BUCKET_CFG0—Offset 110h

Leaky Bucket Configuration Rank 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 110h
**Host Memory Space:** SBREG_BAR + 0x120000 + 110h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 0 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 0, CNTR_HI. |

## 44.5.58 LEAKY_BUCKET_CFG1—Offset 114h

Leaky Bucket Configuration Rank 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 114h
**Host Memory Space:** SBREG_BAR + 0x120000 + 114h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 1 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 1, CNTR_HI. |

## 44.5.59 LEAKY_BUCKET_CFG2—Offset 118h

Leaky Bucket Configuration Rank 2

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 118h
**Host Memory Space:** SBREG_BAR + 0x120000 + 118h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 2 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 2, CNTR_HI. |

## 44.5.60 LEAKY_BUCKET_CFG3—Offset 11Ch

Leaky Bucket Configuration Rank 3

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 11Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 11Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/P | **CNTR_HI:** Specifies the leaky bucket leak rat in CNTR_LO ticks. When the leaky bucket counter counts through CNTR_HI, the correctable error count for rank 3 is decremented by one. |
| 15:0 | 0x0<br>RW/P | **CNTR_LO:** Units for this register are 8192 DRAM clocks. When greater than zero, leaky bucket is enabled. This is the unit for the high order counter for rank 3, CNTR_HI. |

## 44.5.61 FERRNERR—Offset 120h

First Error and Next Error Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 120h
**Host Memory Space:** SBREG_BAR + 0x120000 + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RSV | **Rsvd_FERRNERR_1 (Rsvd_FERRNERR_3):** Reserved |
| 20 | 0x0<br>RO/V/P | **FERRUCEPAT:** First Error Uncorrectable Patrol Scrub Read Error Detected: When set, the first error detected was an uncorrectable (multiple bit) error on the patrol scrub read from memory. |
| 19 | 0x0<br>RO/V/P | **FERRAPE:** First Error Uncorrectable DDR4 Command Address Parity Error: When set, the first error detected was the assertion of the xxDDR3_0_ALERT_N pin indicating that one or more DRAMs detected a Command/Address parity error. |
| 18 | 0x0<br>RO/V/P | **FERRWPE:** First Error Uncorrectable Write Parity Error Detected: When set, the first error detected was an uncorrectable (parity) error on the write data received from the Bunit. |
| 17 | 0x0<br>RO/V/P | **FERRUCE:** First Error Uncorrectable Read Error Detected: When set, the first error detected was an uncorrectable (multiple bit) error on the read data from memory. |
| 16 | 0x0<br>RO/V/P | **FERRSBE:** First Error Correctable Read Error Detected: When set, the first error detected was a single bit error on the read data from memory and the error was corrected. |
| 15:5 | 0x0<br>RSV | **Rsvd_FERRNERR_0 (Rsvd_FERRNERR_1):** Reserved |
| 4 | 0x0<br>RO/V/P | **NERRUCEPAT:** Next Error Uncorrectable Patrol Scrub Read Error: When set, the second error detected was an uncorrectable (mutliple bit) error on the patrol scrub read data from memory. |
| 3 | 0x0<br>RO/V/P | **NERRAPE:** Next Error Uncorrectable DDR4 Command Address Parity Error: When set, the second error detected was the assertion of the xxDDR3_0_ALERT_N pin indicating that one or more DRAMs detected a Command/Address parity error. Note that this bit can never be set as the same time as FERRAPE |
| 2 | 0x0<br>RO/V/P | **NERRWPE:** Next Error Uncorrectable Write Parity Error Detected: When set, the second error detected was an uncorrectable (parity) error on the write data received from the Bunit. |
| 1 | 0x0<br>RO/V/P | **NERRUCE:** Next Error Uncorrectable Read Error Detected: When set, the second error detected was an uncorrectable (multiple bit) error on the read data from memory. |
| 0 | 0x0<br>RO/V/P | **NERRSBE:** Next Error Correctable Read Error Detected: When set, the second error detected was a single bit error on the read data from memory and the error was corrected. |

## 44.5.62 DERRSTS—Offset 124h

Error Status Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 124h
**Host Memory Space:** SBREG_BAR + 0x120000 + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RSV | **Reserved:** Reserved |
| 6 | 0x0<br>RW/1C/V/P | **CAP_ERR_STATE:** Channel Command Address Parity Error State. 1 = CAP error seen, error state entered. All PMI reads will be returned with bad parity. 0 = No CAP error seen. Normal PMI operation. |
| 5 | 0x0<br>RO/V/P | **ALERT:** Current state of the channel ALERT_N pin, captured here as active high: 1 = Error being driven by at least one DRAM device 0 = No error currently detected by any DRAM device |
| 4 | 0x0<br>RW/1C/V/P | **UCEPAT:** Uncorrectable (multiple bit) patrol scrub error: When set, an uncorrectable (multiple bit error was detected on the read data from memory. |
| 3 | 0x0<br>RW/1C/V/P | **APE:** Uncorrectable DDR4 Command/Address Parity Error: When set, an uncorrectable DDR4 command/address parity error was detected by one or more of the DRAMs on channel 0. |
| 2 | 0x0<br>RW/1C/V/P | **WPE:** Uncorrectable (Parity) Write Error: When set, an uncorrectable (parity) error was detected on the write data received from the Bunit. |
| 1 | 0x0<br>RW/1C/V/P | **UCE:** Uncorrectable (Double Bit) Read Error: When set, a double bit error was detected on the read data from memory. |
| 0 | 0x0<br>RW/1C/V/P | **SBE:** Correctable (Single Bit) Read Error: When set, a single bit error was detected on the read data from memory and the error was corrected. |

## 44.5.63 DERRMSKSEV—Offset 128h

Error Mask and Severity

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 128h
**Host Memory Space:** SBREG_BAR + 0x120000 + 128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_2:** Reserved |
| 20 | 0x0<br>RW/P | **MSK_UCEPAT:** Uncorrectable patrol scrub error mask: When set, uncorrectable patrol scrub ECC data errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 19 | 0x0<br>RW/P | **MSK_APE:** Uncorrectable command/address parity error mask: When set, uncorrectable DDR4 command/address parity errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 18 | 0x0<br>RW/P | **MSK_WDBR:** Uncorrectable (parity) write data buffer read error mask: When set, uncorrectable (pairty) write data buffer read errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 17 | 0x0<br>RW/P | **MSK_UCE:** Uncorrectable (multiple bit) error mask: When set, uncorrectable (multiple bit) read and patrol scrub errors will not be logged in DERRESTS[0], DO_SERR_SEV messages will not be sent to the IEH, and derrcnt will not count for this error. |
| 16 | 0x0<br>RW/P | **MSK_SBE:** Correctable (single bit) error mask: When set, correctable (single bit) read and patrol scrub errors will not be logged in DERRESTS[0] and DO_SERR_SEV messages will not be sent to the IEH. |
| 15:5 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_1:** Reserved |
| 4 | 0x0<br>RW/P | **SEV_UCEPAT:** Uncorrectable (Multiple Bit) Patrol Scrub Read Error Severity: When set, UCE errors logged in DERRSTS[5] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 3 | 0x0<br>RW/P | **SEV_APE:** Uncorrectable DDR4 Command/Address Parity Error Severity: When set, APE errors logged in DERRSTS[3] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 2 | 0x0<br>RW/P | **SEV_WDBR:** Uncorrectable (Parity) Write Error Severity: When set, WPE errors logged in DERRSTS[2] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 1 | 0x0<br>RW/P | **SEV_UCE:** Uncorrectable (Multiple Bit) Read Error Severity: When set, UCE errors logged in DERRSTS[1] will be escalated to the global error logic as a fatal error; otherwise it will be escalated as a non-fatal error. |
| 0 | 0x0<br>RSV | **Rsvd_DERRMSKSEV_0:** Reserved |

## 44.5.64 DERRCNTSEL—Offset 12Ch

Uncorrectable Error Count Select

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 12Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 12Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RSV | **Rsvd_DERRCNTSEL_2:** Reserved |
| 4 | 0x0<br>RW/P | **SEL_UCEPAT:** Uncorrectable (Multiple Bit) Patrol Scrub Read Error Count Select: When set, UCE errors logged in DERRSTS[1] will be counted in DERRCNT.UNCCNT1. |
| 3 | 0x0<br>RW/P | **SEL_APE:** Uncorrectable DDR4 Command/Address Parity Error Count Select: When set, APE0 errors logged in DERRSTS[3] will be counted in DERRCNT.UNCCNT0. |
| 2 | 0x0<br>RW/P | **SEL_WPE:** Uncorrectable (Parity) Write Error Count Select: When set, WPE0 errors logged in DERRSTS[2] will be counted in DERRCNT.UNCCNT0. |
| 1 | 0x0<br>RW/P | **SEL_UCE:** Uncorrectable (Multiple Bit) Read Error Count Select: When set, UCE0 errors logged in DERRSTS[1] will be counted in DERRCNT.UNCCNT0. |
| 0 | 0x0<br>RSV | **Rsvd_DERRCNTSEL_0:** Reserved |

## 44.5.65 DERRCNT—Offset 130h

Uncorrectable Error Count

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 130h
**Host Memory Space:** SBREG_BAR + 0x120000 + 130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RSV | **Rsvd_DERRCNT:** Reserved |
| 7:0 | 0x0<br>RW/1C/V/P | **UNCCNT:** Uncorrectable Error Count: The count of uncorrectable errors that have occurred since the last time the count was cleared. The count limit is 128 (80h) and the count is cleared by writing ones to all bits (FFh). The type of uncorrectable errors that are counted is determined by the DERRCNTSEL register. Additionally, if two uncorrectable errors occur in the same clock cycle, only one type will be counted, even if multiple types are selected. |

## 44.5.66 ERRINJCTL—Offset 134h

Error Injection Control

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 134h
**Host Memory Space:** SBREG_BAR + 0x120000 + 134h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/P | **INJECCPAR:** Error Injection ECC/Parity[7:0]: Error injection ECC/parity. Zero bits indicate no injection; one bits indicate that the corresponding read/write ECC or parity bit should be inverted. |
| 23:20 | 0x0<br>RSV | **Rsvd_ERRINJCTL_2:** Reserved |
| 19 | 0x0<br>RW/P | **INJAMSK4:** Error Injection Address Mask[4]: This is the mask bit for error injection address bit 4; address mask bits 36:5 are located in the ERRINJAMSK register. |
| 18 | 0x0<br>RW/P | **INJADDR4:** Error Injection Address[4]: This is error injection address bit 4; address bits 36:5 are located in the ERRINJADDR register. |
| 17 | 0x0<br>RW/P | **INJAMSK3:** Error Injection Address Mask[3]: This is the mask bit for error injection address bit 3; address mask bits 36:5 are located in the ERRINJAMSK register. |
| 16 | 0x0<br>RW/P | **INJADDR3:** Error Injection Address[3]: This is error injection address bit 3; address bits 35:4 are located in the ERRINJADDR register. |
| 15:14 | 0x0<br>RW/P | **INJTRIGSEL:** Error Injection Trigger Selection: 0x = Disable error injection trigger sources 10 = Enable error injection when trigger source 0 (ddr_fsta_trigev[0]) is asserted 11 = Enable error injection when trigger source 1 (ddr_fsta_trigev[1]) is asserted. When setting this field to 1X, the INJEN field should be set to 0 since it will be set automatically when the trigger asserts. |
| 13:12 | 0x0<br>RW/P | **INJCAP:** Command address parity injection per dclk half |
| 11:8 | 0x0<br>RW/P | **INJSRC:** Error Injection Source[3:0]: The selected requestor source ID match against when selecting the transaction for error injection. |
| 7 | 0x0<br>RW/P | **INJPAT:** Inject on patrol scrub request. Address match/mask logic is not used here. Error injection will occur on the next patrol scrub once INJEN is set. |
| 6 | 0x0<br>RW/P | **INJREAD:** Error Injection in the Read Data Path: 0 = Disable 1 = Enabled |
| 5 | 0x0<br>RW/P | **INJWRITE:** Error Injection in the Write Data Path: 0 = Disable 1 = Enabled |
| 4 | 0x0<br>RW/P | **INJECC:** Error Injection in ECC or Parity Path Selection: 0 = Inject in the Data/Parity Path 1 = Inject in the Data/ECC Path |
| 3 | 0x0<br>RW/P | **INJSRCEN:** Error Injection Source Enable: When set, the error injection will be performed only for the specified requestor source. |
| 2 | 0x0<br>RW/P | **INJADREN:** Error Injection Address Enable: When set, the error injection will be performed only for addresses that match the address specified in the ERRINJADDR and ERRINJAMSK registers. |
| 1 | 0x0<br>RW/P | **INJONCE:** Error Injection Once: When set, a single error will be injected after error injection is enabled. Otherwise, error injection will be continuous until disabled. Note that error injection has a minimum temporal granularity of one dunit clock cycle. Thus, if the address match/match bits allow injections on multiple bursts, it is possible to inject up to 4 errors (across the 4 bursts per dunit cycle) even with this bit set. To ensure predictable single error injection INJAMSK3, INJAMSK4 and ERRINJAMSK.INJAMSK[0] must be set to 0. |
| 0 | 0x0<br>RW/V/P | **INJEN:** Error Injection Enable: 0 - Error injection is disabled 1 - Error injection is enabled. This bit is self clearing when inject once is selected and the single error injection has been performed. This bit should be set to 0 when INJTRIGSEL is set to 1X. Upon assertion, the trigger will automatically set the INJEN field. |

## 44.5.67 ERRINJADDR—Offset 138h

Error Injection Address

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 138h
**Host Memory Space:** SBREG_BAR + 0x120000 + 138h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJADDR:** Error Injection PMI Address[36:5] |

## 44.5.68 ERRINJAMSK—Offset 13Ch

Error Injection Address Mask

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 13Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 13Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJAMSK:** Error Injection PMI Address Mask[36:5]: For each bit that is 0 in the address mask, the corresponding PMI address bit will be compared against the same bit in ERRINJADDR. |

## 44.5.69 ERRINJDATA0—Offset 140h

Error Injection Data 0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 140h
**Host Memory Space:** SBREG_BAR + 0x120000 + 140h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJDATA0:** Error Injection Data[31:0]: Zero bits indicate no injection; one bits indicate that the corresponding read/write data bit should be inverted. |

## 44.5.70 ERRINJDATA1—Offset 144h

Error Injection Data 1

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 144h
**Host Memory Space:** SBREG_BAR + 0x120000 + 144h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **INJDATA1:** Error Injection Data[63:32]: Zero bits indicate no injection; one bits indicate that the corresponding read/write data bit should be inverted. |

## 44.5.71 DCS—Offset 170h

DRAM Control and Status

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 170h
**Host Memory Space:** SBREG_BAR + 0x120000 + 170h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RSV | **Rsvd_DCS_5:** Reserved |
| 28 | 0x0<br>RW | **PMICTL:** PMI Control:<br>• 0: Connect to the Bunit<br>• 1: Connect to CPGC<br>When this field is changed, the Dunit will only switch after all previous transactions from the previous unit have been completed. Since the Dunit is a slave to the PMI interface and it cannot prevent the Bunit or CPGC from sendign transactions, the initiatior of this request will have to make sure that the Bunit/CPGC are configured appropriately.<br>Status of this operation from the Dunit perspective is provided in the PMI STAT register. |
| 27:24 | 0x0<br>RSV | **Rsvd_DCS_4:** Reserved |
| 23 | 0x0<br>RW/V/P | **SCRAMBLE_ENABLE:** Data Scrambler Enable:<br>When set to 1 the data lanes (DDR3_DQ[63:0]) and the ECC lanes (DDR3_DQECC[7:0]) will be scrambled, otherwise no scrambling will occur. |
| 22 | 0x0<br>RSV | **Rsvd_DCS_3:** Reserved |
| 21 | 0x0<br>RW/P | **MPRMODE:** Mask Chip-Selects for all commands except read commands and MRS commands |
| 20 | 0x0<br>RW/P | **BLKPHASEB:** Block Phase_B Commands:<br>The Dunit runs at half the frequency of the DRAM clock to save power. In order to issue back-to-back commands on every DRAM clock cycles, the Dunit has to schedule two commands per Dunit clock. Therefore, it processes requests thru two separate arbitration paths labeled as Phase_A and Phase_B. Setting this bit to a 1 will disable Phase_B completely, and all memory requests will be processed thru Phase_A arbitration path. With only Phase_A, the DRAM devices will never see back-to-back commands. The best it will see is a command every other DRAM clock cycle.<br>• 0 - No block<br>• 1 - Block |
| 19:18 | 0x0<br>RSV | **Rsvd_DCS_2:** Reserved |
| 17:16 | 0x0<br>RW/P | **ECCLANESWAP:** ECC Byte Lane Swap: This feature is not supported in 32 bit mode and this field must be set to 0.<br>• 0h - Normal mode<br>• 1h - Swap ECC and Data Byte 0<br>• 2h - Reserved<br>• 3h - Reserved |
| 15:13 | 0x0<br>RSV | **Rsvd_DCS_1:** Reserved |
| 12 | 0x0<br>RW/P | **ODTMODE:** ODT Control Mode:<br>• 0 - Dunit auto controls the ODT pins based on DRAM Write transactions.<br>• 1 - The value of ODTVAL above directly controls the ODT pins. |
| 11:8 | 0x0<br>RW/P | **ODTVAL:** ODT Control Value: Each bit of ODTVAL directly controls an ODT pin, when ODTMODE is set to 1. Used only during init flow by BIOS. |
| 7:5 | 0x0<br>RSV | **Rsvd_DCS_0:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW/P | **CKEMODE:** CKE Control Mode:<br>• 0 - Dunit auto controls the CKE pins based on Power-Down and Self Refresh entry and exit.<br>• 1 - The value of CKEVAL above directly controls the CKE pins. |
| 3:0 | 0x0<br>RW/P | **CKEVAL:** CKE Control Value: Each bit of CKEVAL directly controls a CKE pin, when CKEMODE is set to 1. Used only during init flow by BIOS. |

## 44.5.72 DECCCTRL—Offset 180h

Dunit ECC Control

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 180h
**Host Memory Space:** SBREG_BAR + 0x120000 + 180h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:26 | 0x0<br>RSV | **Rsvd_DECCCTRL_1:** Reserved |
| 25 | 0x0<br>RW/P | **RD2CLRACC:** Read to Clear C[1:0]SBEACC Registers:<br>• 0 = C[1:0]SBEACC Registers are cleared when written<br>• 1 = C[1:0]SBEACC Registers are cleared when written or read |
| 24:18 | 0x0<br>RSV | **Rsvd_DECCCTRL_0:** Reserved |
| 17 | 0x0<br>RW/P | **ENCBGEN:** Enable Generation of ECC Check Bits:<br>• 0 - Disable generation of ECC Check Bits<br>• 1 - Enable generation of ECC Check Bits<br>ECC is generated for both the Bunit and the CPGC |
| 16 | 0x0<br>RW/P | **ECCOVREN:** ECC Check Bits Override Enable:<br>• 0 - Normal ECC is generated.<br>• 1 - ECC override enabled, the value in ECCOVR will be used for the ECC code. |
| 15:8 | 0x0<br>RW/P | **ECCOVR:** ECC Check Bits Override Value: The ECC override code to be used on writes if enabled by ECCOVREN. |
| 7:5 | 0x0<br>RW/P | **UCE_FILTER:** Uncorrectable Error Filter:<br>• 000: Normal mode; All 8 byte ECC checks/corrections are performed independently, no anti-aliasing filters are applied.<br>• 011: An uncorrectable error that is detected in any 8 byte ECC check will cause all four parallel 8 byte ECC checks to be treated as uncorrectable errors. |
| 4 | 0x0<br>RW/P | **DPAREN:** Data Parity Enable: When set, data path parity is enabled. The write data parity is checked at the point when ECC is generated and read data parity is generated at the point that ECC is checked. |
| 3 | 0x0<br>RW/P | **CAPERREN:** Enable Command Address Parity Error Detection: When set, command address parity errors are detected and signalled to the Bunit. |
| 2 | 0x0<br>RW/P | **DERREN:** Enable Uncorrectable Error Notification: When set, uncorrectable ECC errors are signalled to the Bunit and patrol scrub. |
| 1 | 0x0<br>RW/P | **SERREN:** Enable Correctable Error Notification: When set, correctable ECC errors are signalled to the Bunit and patrol scrub which indicates that the read data needs to be written back to memory. This causes the DRAM cell with the single bit error to be scrubbed. |
| 0 | 0x0<br>RW/V/P | **ECCEN:** ECC Enable: When set, single bit error correction and double bit error detection (SEC/DED ECC) is enabled. The ECC is an 8 bit protection code for every 64 bits of data stored in DRAM. ECC is generated on writes and checked on reads, requiring a 72 bit DDR3 data bus width. |

## 44.5.73 LEAKY_BUCKET_CNTR_UPPER—Offset 18Ch

Leaky Bucket Counter Upper

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 18Ch
**Host Memory Space:** SBREG_BAR + 0x120000 + 18Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/P | **CNTR_UPPER_RANK3:** Rank 3 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK3 + 1)* (LEAKY_BUCKET_CFG3.CNTR_HI + 1) * (LEAKY_BUCKET_CFG3.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 23:16 | 0x0<br>RW/P | **CNTR_UPPER_RANK2:** Rank 2 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK2 + 1)* (LEAKY_BUCKET_CFG2.CNTR_HI + 1) * (LEAKY_BUCKET_CFG2.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 15:8 | 0x0<br>RW/P | **CNTR_UPPER_RANK1:** Rank 1 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK1 + 1)* (LEAKY_BUCKET_CFG1.CNTR_HI + 1) * (LEAKY_BUCKET_CFG1.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |
| 7:0 | 0x0<br>RW/P | **CNTR_UPPER_RANK0:** Rank 0 CNTR_UPPER configuration.<br>Total Leak Rate = (CNTR_UPPER_RANK0 + 1)* (LEAKY_BUCKET_CFG0.CNTR_HI + 1) * (LEAKY_BUCKET_CFG0.CNTR_LO + 1) * 8192 DRAM clocks. Setting CNTR_LO to zero will disable the leaky bucket. |

## 44.5.74 DPATROL_SCRUB_TMR—Offset 194h

Patrol Scrub Timer Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x12] + 194h
**Host Memory Space:** SBREG_BAR + 0x120000 + 194h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RW/P | **PATSCRTIMER:** Patrol scrub interval timer in DCLKs (DRAM cmd clk divided by 2). A new patrol scrub is scheduled every time the timer ticks to the value in this register. Any non zero value is valid |

## 44.5.75 ICLK_DWORD8—Port 99h, Offset 220h

Flex Clock 0 and Flex Clock 1 Frequency Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x99] + 220h
**Host Memory Space:** SBREG_BAR + 0x990000 + 220h

**Default:** 0004_2222h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RSV | **Reserved** |
| 13 | 0x1<br>RW/V | **sel_flex_base1:** Selects the frequency of flex_base1:<br>0=200 MHz for flex1 clock<br>1=96 MHz for flex1 clock |
| 12 | 0x0<br>RSV | **Reserved** |
| 11:8 | 0x2<br>RW/V | **flex_ck_div1:** flex clock 1 is flex_base1/flex_ck_div1. |
| 7:6 | 0x0<br>RSV | **Reserved** |
| 5 | 0x1<br>RW/V | **sel_flex_base0** Selects the frequency of flex_base0:<br>0=200 MHz for flex1 clock<br>1=96 MHz for flex1 clock |
| 4 | 0x0<br>RSV | **Reserved** |
| 3:0 | 0x2<br>RW/V | **flex_ck_div0:** flex clock 0 is flex_base0/flex_ck_div0. |

§ §

# 45 Global and Local Error Registers (GLMREG) - B0, D4, F0

## 45.1 Introduction and Index

The host-accessible registers for the Global and Local Error Registers (GLMREG) are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 45.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 4 (decimal), Function 0. The offset addresses are listed.

**Table 45-1. Summary of PCI Configuration Registers—0/4/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "GLREG Vendor ID (GLREG_VID)—Offset 0h" on page 1483 |
| 4 | 2 | 0000 | "GLREG PCI Command (GLREG_PCICMD)—Offset 4h" on page 1484 |
| 6 | 2 | 0010 | "GLREG PCI Status (GLREG_PCISTS)—Offset 6h" on page 1485 |
| 9 | 3 | 00060000 | "GLREG Class Code (GLREG_CC)—Offset 9h" on page 1486 |
| C | 1 | 00 | "GLREG Cacheline Size (GLREG_CLS)—Offset Ch" on page 1486 |
| 2C | 2 | 8086 | "GLREG Subsystem Vendor ID (GLREG_SVID)—Offset 2Ch" on page 1487 |
| 34 | 1 | 40 | "GLREG Capabilities Pointer (GLREG_CAPPTR)—Offset 34h" on page 1487 |
| 3D | 3 | 00 | "GLREG Interrupt Pin (GLREG_INTP)—Offset 3Dh" on page 1488 |
| 40 | 2 | 0010 | "GLREG PCI Express Capability List (GLREG_EXPCAPLST)—Offset 40h" on page 1488 |
| 42 | 2 | 0092 | "GLREG PCI Express Capability List (GLREG_EXPCAPLST)—Offset 40h" on page 1488 |
| 48 | 2 | 0000 | "GLREG Device Control (GLREG_DEVCTL)—Offset 48h" on page 1490 |
| 4A | 1 | 00 | "GLREG Device Status (GLREG_DEVSTS)—Offset 4Ah" on page 1491 |
| 200 | 4 | 00000000 | "GLREG Global Correctable Error Status (GLREG_GCOERRSTS)—Offset 200h" on page 1492 |
| 204 | 4 | 00000000 | "GLREG Global Non-Fatal Error Status (GLREG_GNFERRSTS)—Offset 204h" on page 1494 |
| 208 | 4 | 00000000 | "GLREG Global Fatal Error Status (GLREG_GFAERRSTS)—Offset 208h" on page 1496 |
| 20C | 4 | 00000000 | "GLREG Global Error Mask (GLREG_GERRMSK)—Offset 20Ch" on page 1498 |
| 228 | 8 | 0000000000000000 | "GLREG Global Error Timer (GLREG_GTIME)—Offset 228h" on page 1500 |
| 230 | 8 | 0000000000000000 | "GLREG Global Correctable FERR Error Time Stamp (GLREG_GCOFERRTIME)—Offset 230h" on page 1500 |
| 238 | 8 | 0000000000000000 | "GLREG Global Nonfatal FERR Error Time Stamp (GLREG_GNFFERRTIME)—Offset 238h" on page 1501 |
| 240 | 8 | 0000000000000000 | "GLREG Global Fatal FERR Error Time Stamp (GLREG_GFAFERRTIME)—Offset 240h" on page 1501 |
| 248 | 4 | 00000000 | "GLREG Global System Event Status (GLREG_GSYSEVTSTS)—Offset 248h" on page 1502 |
| 24C | 4 | 00000000 | "GLREG Global System Event Mask (GLREG_GSYSEVTMSK)—Offset 24Ch" on page 1502 |
| 250 | 4 | 00000025 | "GLREG Global System Event Map (GLREG_GSYSEVTMAP)—Offset 250h" on page 1503 |
| 254 | 4 | 00000025 | "GLREG Error Pin Control (GLREG_ERRPINCTRL)—Offset 254h" on page 1504 |
| 258 | 4 | 00000000 | "GLREG Error Pin Status (GLREG_ERRPINSTS)—Offset 258h" on page 1504 |
| 25C | 4 | 00000000 | "GLREG Error Pin Data (GLREG_ERRPINDATA)—Offset 25Ch" on page 1505 |
| 294 | 1 | 00 | "GLREG Uncorrectable Error Counter (GLREG_LUNCERRCNT)—Offset 294h" on page 1506 |

## 45.2 Registers in Configuration Space

### 45.2.1 GLREG Vendor ID (GLREG_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **VID:** This field identifies Intel as the manufacturer of the device. |

## 45.2.2 GLREG PCI Command (GLREG_PCICMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved:** Reserved. |
| 10 | 0x0<br>RO | **INTxD:** This bit controls the ability of the PCI-Express Function to generate INTx interrupt message. When set, functions are prevented from asserting INTx interrupt messages.<br>Any INTx emulation interrupts already asserted by the function must be deserted when this bit is set by generating a Deassert_INTx message(s). |
| 9 | 0x0<br>RO | **FBE:** Not applicable to PCI-Express. Hardwired to 0. |
| 8 | 0x0<br>RO | **SEE:** When set, this bit enables reporting of Non-Fatal and Fatal Errors detected by the Function to the Root Complex. ERR_COR messages are not affected by this bit.<br>Errors are reported when enabled either through this bit or through the PCI Express-specific bits in the Device Control Register (GLREG_DEVCTL). |
| 7 | 0x0<br>RO | **WCC:** Not applicable to PCI-Express. Hardwired to 0. |
| 6 | 0x0<br>RO | **PERE:** This bit controls the setting of the master data parity error bit in the Status Register (GLREG_PCISTS) in response to a parity error received on the PCI Express interface (poisoned TLP). |
| 5 | 0x0<br>RO | **VGA_PSE:** Not applicable to PCI-Express. Hardwired to 0. |
| 4 | 0x0<br>RO | **MWIE:** Not applicable to PCI-Express. Hardwired to 0. |
| 3 | 0x0<br>RO | **SCE:** Not applicable to PCI-Express. Hardwired to 0. |
| 2 | 0x0<br>RO | **BME:** This bit controls the ability of the Function to issue Memory and I/O read or write requests, and the ability of Root or Switch port to forward memory and I/O read or write requests in the upstream direction. |
| 1 | 0x0<br>RO | **MSE:** This bit controls the function's response to Memory Space accesses. |
| 0 | 0x0<br>RO | **IOSE:** This bit controls the function's response to IO Space accesses. |

## 45.2.3 GLREG PCI Status (GLREG_PCISTS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **DPE:** This bit is set when a poisoned TLP is received.<br>This bit is set even when the parity error response enable bit (bit[6] of the PCICMD Register (GLREG_PCICMD) is not set. |
| 14 | 0x0<br>RO | **SSE:** This bit is set when ERR_FATAL or ERR_NONFATAL messages are sent to the root complex event collector and the SERR enable bit in the PCICMD Register (GLREG_PCICMD) is set. |
| 13 | 0x0<br>RO | **RMA:** This bit is set when the requester receives a completion with an UR completion status. |
| 12 | 0x0<br>RO | **RTA:** This bit is set when a requester receives a CA completions status. |
| 11 | 0x0<br>RO | **STA:** This bit is set when the switch generates a completion packet with Completer Abort (CA) status is generated by its primary side. |
| 10:9 | 0x0<br>RO | **Reserved:** Reserved |
| 8 | 0x0<br>RO | **MDPD:** This bit is set by a requester if the parity error response enable bit (PERE) in the Command Register (GLREG_PCICMD) is set and either of the following two conditions occur:<br>• Requester receives a completion marked poisoned.<br>• Requester poisons a write requests.<br>If the parity error bit is 0b, this bit is never set. |
| 7 | 0x0<br>RO | **Reserved:** Reserved |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **Reserved:** Reserved |
| 4 | 0x1<br>RO | **CAPE:** This bit indicates the presence of an Extended capabilities list items.<br>This bit is hardwired to 1b. |
| 3 | 0x0<br>RO | **INTS:** When set, this bit indicates that an INTx emulation interrupt is pending internally in this function. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 45.2.4 GLREG Class Code (GLREG_CC)—Offset 9h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 9h

**Default:** 00060000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x6<br>RO | **BC:** This function is hardwired to 06h indicating a Host bridge. |
| 15:8 | 0x0<br>RO | **SC:** This function is hardwired to 00h, indicating a Host Bridge. |
| 7:0 | 0x0<br>RO | **RLPI:** Set to 00h for all non-APIC devices. |

## 45.2.5 GLREG Cacheline Size (GLREG_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:4, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **CLS:** These bits specify the system cache-line size in units of DWords.<br>This field is implemented by PCI Express devices but has no effect on device behavior. |

## 45.2.6 GLREG Subsystem Vendor ID (GLREG_SVID)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 2Ch

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RW/O | **SVID:** This field identifies Intel as the manufacturer of the device. |

## 45.2.7 GLREG Capabilities Pointer (GLREG_CAPPTR)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:4, F:0] + 34h

**Default:** 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x40<br>RO | **CPTR:** Contains the offset of the first item in the list of capabilities. (GLREG_EXPCAPLST) |

## 45.2.8 GLREG Interrupt Pin (GLREG_INTP)—Offset 3Dh

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:4, F:0] + 3Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **INTP:** This register tells which interrupt pin the function uses.<br>• 01h: Generate INTA<br>• 02h: Generate INTB<br>• 03h: Generate INTC<br>• 04h: Generate INTD<br>• Others: Reserved<br>• Devices (or device functions) that do not use an interrupt pin must put a 0 in this register.<br>Not applicable since this device does not generate any interrupts on its own. |

## 45.2.9 GLREG PCI Express Capability List (GLREG_EXPCAPLST)—Offset 40h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 40h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **NP:** Contains the offset of the next item in the capabilities list. Indicates that this is the last capability in the list. |
| 7:0 | 0x10<br>RO | **CAPID:** Identifies the function as PCI Express capable. |

## 45.2.10 GLREG PCI Express Capabilities (GLREG_EXPCAP)—Offset 42h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 42h

**Default:** 0092h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13:9 | 0x0<br>RO | **IMN:** This field indicates the interrupt message number that is generated from the PCI Express port. |
| 8 | 0x0<br>RO | **SI:** Indicates the PCI Express link associated with this port is connected to a slot. |
| 7:4 | 0x9<br>RO | **DT:** 1001b Root Complex Integrated Endpoint (RCEC). |
| 3:0 | 0x2<br>RO | **VN:** These bits indicate the version number of the PCI Express capability structure. |

## 45.2.11 GLREG Device Control (GLREG_DEVCTL)—Offset 48h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:4, F:0] + 48h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RO | **MRRS:** This field sets the maximum Read Requests size for the function as a requester. The Function must not generate read requests with size exceeding the set value.<br>• 000b: 128 bytes maximum Read Request size<br>• 001b: 256 bytes maximum Read Request size<br>• 010b: 512 bytes maximum Read Request size<br>• 011b: 1024 bytes maximum Read Request size<br>• 100b: 2048 bytes maximum Read Request size<br>• 101b: 4096 bytes maximum Read Request size<br>• Others: Reserved<br>Functions that do not generate Read Requests larger than 128B and functions that do not generate Read Requests on their own behalf are permitted to implement this field as Read Only (RO) with a value of 000b. |
| 11 | 0x0<br>RO | **ENOSNP:** This bit is permitted to be hardwired to 0b if a function would never set the No Snoop attribute in transactions it initiates. |
| 10 | 0x0<br>RO | **AUXPME:** Not supported |
| 9 | 0x0<br>RO | **PFE:** Functions that do not implement this capability hardware this bit to 0b. |
| 8 | 0x0<br>RO | **ETFE:** Functions that do not implement this capability hardwire this bit to 0b. |
| 7:5 | 0x0<br>RW | **MPS:** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as larger as the set value. As a Transmitter, the function must not generate TLPs exceeding the set value.<br>• 000b: 128 bytes maximum payload size<br>• 001b: 256 bytes maximum payload size<br>• 010b: 512 bytes maximum payload size (Unsupported)<br>• 011b: 1024 bytes maximum payload size (Unsupported)<br>• 100b: 2048 bytes maximum payload size (Unsupported)<br>• 101b: 4096 bytes maximum payload size (Unsupported)<br>• Others: Reserved |
| 4 | 0x0<br>RO | **ENRO:** A function is permitted to hardwire this bit to 0b if it never sets the Relaxed ordering attribute in transactions it initiates as a requester. |
| 3 | 0x0<br>RO | **URRE:** This bit controls the enabling of ERR_CORR, ERR_NONFATAL or ERR_FATAL messages on PCI Express for reporting Unsupported Request (UR) errors. |
| 2 | 0x0<br>RO | **FERE:** Generation of the ERR_FATAL message is disabled. |
| 1 | 0x0<br>RO | **NFERE:** Generation of the ERR_NONFATAL message is disabled. |
| 0 | 0x0<br>RO | **CERE:** Generation of the ERR_CORR message is disabled. |

## 45.2.12 GLREG Device Status (GLREG_DEVSTS)—Offset 4Ah

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:4, F:0] + 4Ah

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **Reserved:** Reserved |
| 4 | 0x0<br>RO | **APD:** Auxiliary Power is not supported. |
| 3 | 0x0<br>RO | **URD:** This bit indicates that the device received an Unsupported Request (UR). |
| 2 | 0x0<br>RO | **FED:** This bit indicates status of Fatal Errors detected. |
| 1 | 0x0<br>RO | **NFED:** This bit indicates status of Non-Fatal Errors detected. |
| 0 | 0x0<br>RO | **CED:** This bit indicates status of Correctable Errors detected. |

## 45.2.13 GLREG Global Correctable Error Status (GLREG_GCOERRSTS)—Offset 200h

This register indicates that a Correctable Error is reported to the global error logic. An individual error status bit that is set indicates that a particular local device or RCEC has detected an error.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 200h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0 RW/1C/V/P | **Reserved:** Reserved. |
| 23 | 0x0 RW/1C/V/P | **NPK_CO_STS:** NPK Error Status - Not used by the SoC |
| 22 | 0x0 RW/1C/V/P | **SPIESPI_CO_STS:** SPIESPI Error Status - Not used by the SoC |
| 21 | 0x0 RW/1C/V/P | **HSUART_CO_STS:** HSUART Error Status - Not used by the SoC |
| 20 | 0x0 RW/1C/V/P | **SMBUS_CO_STS:** SMBUS Error Status - Not used by the SoC |
| 19 | 0x0 RW/1C/V/P | **LPC_CO_STS:** LPC Error Status - Not used by the SoC |
| 18 | 0x0 RW/1C/V/P | **USB_CO_STS:** USB Error Status - Not used by the SoC |
| 17 | 0x0 RW/1C/V/P | **SATA1_CO_STS:** SATA1 Error Status - Not used by the SoC |
| 16 | 0x0 RW/1C/V/P | **SATA0_CO_STS:** SATA0 Error Status - Not used by the SoC |
| 15 | 0x0 RW/1C/V/P | **VRP2_CO_STS:** |
| 14 | 0x0 RW/1C/V/P | **VRP1_CO_STS:** |
| 13 | 0x0 RW/1C/V/P | **VRP0_CO_STS:** |
| 12 | 0x0 RW/1C/V/P | **PCIeRP7_CO_STS:** |
| 11 | 0x0 RW/1C/V/P | **PCIeRP6_CO_STS:** |
| 10 | 0x0 RW/1C/V/P | **PCIeRP5_CO_STS:** |
| 9 | 0x0 RW/1C/V/P | **PCIeRP4_CO_STS:** |
| 8 | 0x0 RW/1C/V/P | **PCIeRP3_CO_STS:** |
| 7 | 0x0 RW/1C/V/P | **PCIeRP2_CO_STS:** |
| 6 | 0x0 RW/1C/V/P | **PCIeRP1_CO_STS:** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/1C/V/P | **PCIeRP0_CO_STS:** |
| 4 | 0x0<br>RW/1C/V/P | **Dunit1_CO_STS:** |
| 3 | 0x0<br>RW/1C/V/P | **Dunit0_CO_STS:** |
| 2 | 0x0<br>RW/1C/V/P | **Bunit_CO_STS:** |
| 1 | 0x0<br>RW/1C/V/P | **Aunit_CO_STS:** |
| 0 | 0x0<br>RW/1C/V/P | **RCEC_CO_STS:** |

## 45.2.14 GLREG Global Non-Fatal Error Status (GLREG_GNFERRSTS)—Offset 204h

This register indicates that a Non-Fatal Error is reported to the global error logic. An individual error status bit that is set indicates that a particular local device or RCEC has detected an error.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 204h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0 RW/1C/V/P | **Reserved:** Reserved. |
| 23 | 0x0 RW/1C/V/P | **NPK_NF_STS:** NPK Error Status - Not used by the SoC |
| 22 | 0x0 RW/1C/V/P | **SPIESPI_NF_STS:** SPIESPI Error Status - Not used by the SoC |
| 21 | 0x0 RW/1C/V/P | **HSUART_NF_STS:** Reserved. |
| 20 | 0x0 RW/1C/V/P | **SMBUS_NF_STS:** SMBUS Error Status - Not used by the SoC |
| 19 | 0x0 RW/1C/V/P | **LPC_NF_STS:** LPC Error Status - Not used by the SoC |
| 18 | 0x0 RW/1C/V/P | **USB_NF_STS:** USB Error Status - Not used by the SoC |
| 17 | 0x0 RW/1C/V/P | **SATA1_NF_STS:** SATA1 Error Status - Not used by the SoC |
| 16 | 0x0 RW/1C/V/P | **SATA0_NF_STS:** SATA0 Error Status - Not used by the SoC |
| 15 | 0x0 RW/1C/V/P | **VRP2_NF_STS:** |
| 14 | 0x0 RW/1C/V/P | **VRP1_NF_STS:** |
| 13 | 0x0 RW/1C/V/P | **VRP0_NF_STS:** |
| 12 | 0x0 RW/1C/V/P | **PCIeRP7_NF_STS:** |
| 11 | 0x0 RW/1C/V/P | **PCIeRP6_NF_STS:** |
| 10 | 0x0 RW/1C/V/P | **PCIeRP5_NF_STS:** |
| 9 | 0x0 RW/1C/V/P | **PCIeRP4_NF_STS:** |
| 8 | 0x0 RW/1C/V/P | **PCIeRP3_NF_STS:** |
| 7 | 0x0 RW/1C/V/P | **PCIeRP2_NF_STS:** |
| 6 | 0x0 RW/1C/V/P | **PCIeRP1_NF_STS:** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/1C/V/P | **PCIeRP0_NF_STS:** |
| 4 | 0x0<br>RW/1C/V/P | **Dunit1_NF_STS:** |
| 3 | 0x0<br>RW/1C/V/P | **Dunit0_NF_STS:** |
| 2 | 0x0<br>RW/1C/V/P | **Bunit_NF_STS:** |
| 1 | 0x0<br>RW/1C/V/P | **Aunit_NF_STS:** |
| 0 | 0x0<br>RW/1C/V/P | **RCEC_NF_STS:** |

## 45.2.15 GLREG Global Fatal Error Status (GLREG_GFAERRSTS)—Offset 208h

This register indicates that a Fatal Error is reported to the global error logic. An individual error status bit that is set indicates that a particular local device or RCEC has detected an error.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 208h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0 RW/1C/V/P | **Reserved:** Reserved. |
| 23 | 0x0 RW/1C/V/P | **NPK_FA_STS:** |
| 22 | 0x0 RW/1C/V/P | **SPIESPI_FA_STS:** |
| 21 | 0x0 RW/1C/V/P | **HSUART_FA_STS:** |
| 20 | 0x0 RW/1C/V/P | **SMBUS_FA_STS:** |
| 19 | 0x0 RW/1C/V/P | **LPC_FA_STS:** |
| 18 | 0x0 RW/1C/V/P | **USB_FA_STS:** |
| 17 | 0x0 RW/1C/V/P | **SATA1_FA_STS:** |
| 16 | 0x0 RW/1C/V/P | **SATA0_FA_STS:** |
| 15 | 0x0 RW/1C/V/P | **VRP2_FA_STS:** |
| 14 | 0x0 RW/1C/V/P | **VRP1_FA_STS:** |
| 13 | 0x0 RW/1C/V/P | **VRP0_FA_STS:** |
| 12 | 0x0 RW/1C/V/P | **PCIeRP7_FA_STS:** |
| 11 | 0x0 RW/1C/V/P | **PCIeRP6_FA_STS:** |
| 10 | 0x0 RW/1C/V/P | **PCIeRP5_FA_STS:** |
| 9 | 0x0 RW/1C/V/P | **PCIeRP4_FA_STS:** |
| 8 | 0x0 RW/1C/V/P | **PCIeRP3_FA_STS:** |
| 7 | 0x0 RW/1C/V/P | **PCIeRP2_FA_STS:** |
| 6 | 0x0 RW/1C/V/P | **PCIeRP1_FA_STS:** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/1C/V/P | **PCIeRP0_FA_STS:** |
| 4 | 0x0<br>RW/1C/V/P | **Dunit1_FA_STS:** |
| 3 | 0x0<br>RW/1C/V/P | **Dunit0_FA_STS:** |
| 2 | 0x0<br>RW/1C/V/P | **Bunit_FA_STS:** |
| 1 | 0x0<br>RW/1C/V/P | **Aunit_FA_STS:** |
| 0 | 0x0<br>RW/1C/V/P | **RCEC_FA_STS:** |

## 45.2.16 GLREG Global Error Mask (GLREG_GERRMSK)—Offset 20Ch

This register masks the reporting of errors detected by local devices. An individual error mask bit that is set masks error reporting of the particular local device, software may set or clear the mask bit. This register is sticky and can only be reset by PWRGOOD. Global Error Mask register masks errors reported from the local devices or RCEC to the global register. If the error reporting is masked in this register, all errors from the corresponding local device or RCEC will not set any of the global error status bits.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 20Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/P | **Reserved:** Reserved. |
| 23 | 0x0<br>RW/P | **NPK_MSK:**<br>• 1=mask<br>• 0=unmask |
| 22 | 0x0<br>RW/P | **SPIESPI_MSK:**<br>• 1=mask<br>• 0=unmask |
| 21 | 0x0<br>RW/P | **HSUART_MSK:**<br>• 1=mask<br>• 0=unmask |
| 20 | 0x0<br>RW/P | **SMBUS_MSK:**<br>• 1=mask<br>• 0=unmask |
| 19 | 0x0<br>RW/P | **LPC_MSK:**<br>• 1=mask<br>• 0=unmask |
| 18 | 0x0<br>RW/P | **USB_MSK:**<br>• 1=mask<br>• 0=unmask |
| 17 | 0x0<br>RW/P | **SATA1_MSK:**<br>• 1=mask<br>• 0=unmask |
| 16 | 0x0<br>RW/P | **SATA0_MSK:**<br>• 1=mask<br>• 0=unmask |
| 15 | 0x0<br>RW/P | **VRP2_MSK:**<br>• 1=mask<br>• 0=unmask |
| 14 | 0x0<br>RW/P | **VRP1_MSK:**<br>• 1=mask<br>• 0=unmask |
| 13 | 0x0<br>RW/P | **VRP0_MSK:**<br>• 1=mask<br>• 0=unmask |
| 12 | 0x0<br>RW/P | **PCIeRP7_MSK:**<br>• 1=mask<br>• 0=unmask |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x0<br>RW/P | **PCIeRP6_MSK:**<br>• 1=mask<br>• 0=unmask |
| 10 | 0x0<br>RW/P | **PCIeRP5_MSK:**<br>• 1=mask<br>• 0=unmask |
| 9 | 0x0<br>RW/P | **PCIeRP4_MSK:**<br>• 1=mask<br>• 0=unmask |
| 8 | 0x0<br>RW/P | **PCIeRP3_MSK:**<br>• 1=mask<br>• 0=unmask |
| 7 | 0x0<br>RW/P | **PCIeRP2_MSK:**<br>• 1=mask<br>• 0=unmask |
| 6 | 0x0<br>RW/P | **PCIeRP1_MSK:**<br>• 1=mask<br>• 0=unmask |
| 5 | 0x0<br>RW/P | **PCIeRP0_MSK:**<br>• 1=mask<br>• 0=unmask |
| 4 | 0x0<br>RW/P | **Dunit1_MSK:**<br>• 1=mask<br>• 0=unmask |
| 3 | 0x0<br>RW/P | **Dunit0_MSK:**<br>• 1=mask<br>• 0=unmask |
| 2 | 0x0<br>RW/P | **Bunit_MSK:**<br>• 1=mask<br>• 0=unmask |
| 1 | 0x0<br>RW/P | **Aunit_MSK:**<br>• 1=mask<br>• 0=unmask |
| 0 | 0x0<br>RW/P | **RCEC_MSK:**<br>• 1=mask<br>• 0=unmask |

## 45.2.17 GLREG Global Error Timer (GLREG_GTIME)—Offset 228h

Global Error Timer register is a free running 64-bit counter and will indicate the current value of the 64-bit counter. This counter is reset to 0 by PWRGOOD. Once out of PWRGOOD reset, the counter begins to run.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:4, F:0] + 228h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V/P | **GTIME_Value:** This is the 64-bit free running counter. |

## 45.2.18 GLREG Global Correctable FERR Error Time Stamp (GLREG_GCOFERRTIME)—Offset 230h

The time stamp register logs the 64-bit free running counter when the first Correctable Error was logged.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:4, F:0] + 230h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V/P | **G_COFERR_TIME:** The time stamp register logs the 64-bit free running counter when the first Correctable Error was logged. |

## 45.2.19 GLREG Global Nonfatal FERR Error Time Stamp (GLREG_GNFFERRTIME)—Offset 238h

The time stamp register logs the 64-bit free running counter when the first Nonfatal Error was logged.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:4, F:0] + 238h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V/P | **G_NFFERR_TIME:** The time stamp register logs the 64-bit free running counter when the first Non-Fatal Error was logged. |

## 45.2.20 GLREG Global Fatal FERR Error Time Stamp (GLREG_GFAFERRTIME)—Offset 240h

The time stamp register logs the 64-bit free running counter when the first Fatal Error was logged.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:4, F:0] + 240h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:0 | 0x0<br>RO/V/P | **G_FAFERR_TIME:** The time stamp register logs the 64-bit free running counter when the first Fatal Error was logged. |

## 45.2.21 GLREG Global System Event Status (GLREG_GSYSEVTSTS)—Offset 248h

This register indicates the error severity signaled by the global error logic. Setting of an individual error status bit indicates that the corresponding error severity has been detected.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 248h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Reserved:** Reserved. |
| 2 | 0x0<br>RO/V/P | **FA_SYSEV_STS:** A Fatal Error was detected when set |
| 1 | 0x0<br>RO/V/P | **NF_SYSEV_STS:** A Nonfatal Error was detected when set |
| 0 | 0x0<br>RO/V/P | **CO_SYSEV_STS:** A Correctable Error was detected when set |

## 45.2.22 GLREG Global System Event Mask (GLREG_GSYSEVTMSK)—Offset 24Ch

The System Event Mask register masks the reporting the errors indicated by the system event status register. When set, the error severity does not cause the generation of the system event. When cleared, detection of the error severity generates system event(s) according to System Event Map Register (SYSEVMAP)

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 24Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Reserved:** Reserved. |
| 2 | 0x0<br>RW/P | **FA_SYSEV_M:** Fatal Error System Event Mask<br>• 0 - Unmask system event reporting of the Fatal Error<br>• 1 - Mask system event reporting of the Fatal Error |
| 1 | 0x0<br>RW/P | **NF_SYSEV_M:** Nonfatal Error System Event Mask<br>• 0 - Unmask system event reporting of the nonfatal error<br>• 1 - Mask system event reporting of the Nonfatal Error |
| 0 | 0x0<br>RW/P | **CO_SYSEV_M:** Correctable Error System Event Mask<br>• 0 - Unmask system event reporting of the Correctable Error<br>• 1 - Mask system event reporting of the Correctable Error |

## 45.2.23 GLREG Global System Event Map (GLREG_GSYSEVTMAP)—Offset 250h

This register maps the error severity detected to one of the system events. When an error is detected, its corresponding error severity determines which system event to generate according to this register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 250h

**Default:** 00000025h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved:** Reserved. |
| 5:4 | 0x2<br>RW/P | **FA_SYSEV_MAP:**<br>• 11: Reserved<br>• 10: Generate NMI<br>• 01: Generate SMI<br>• 00: No system event generation<br>***Note:*** NMIE.NMI=1 and NSC.SERR#_NMI_enable=1 to generate an NMI. |
| 3:2 | 0x1<br>RW/P | **NF_SYSEV_MAP:**<br>• 11: Reserved<br>• 10: Generate NMI<br>• 01: Generate SMI<br>• 00: No system event generation<br>***Note:*** NMIE.NMI=1 and NSC.SERR#_NMI_enable=1 to generate an NMI. |
| 1:0 | 0x1<br>RW/P | **CO_SYSEV_MAP:**<br>• 11: Reserved<br>• 10: Generate NMI<br>• 01: Generate SMI<br>• 00: No system event generation<br>***Note:*** NMIE.NMI=1 and NSC.SERR#_NMI_enable=1 to generate an NMI. |

## 45.2.24 GLREG Error Pin Control (GLREG_ERRPINCTRL)—Offset 254h

This register provides the option to configure an error pin to either as a special purpose error pin which is asserted based on the detected error severity, or as a general purpose output which is asserted based on the value in the ERRPINDAT. The assertion of the error pins can also be completely disabled by this register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 254h

**Default:** 00000025h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved:** Reserved. |
| 5:4 | 0x2<br>RW/P | **ERR2_ENABLE:**<br>• 11: Reserved.<br>• 10: Assert Error Pin when Fatal Error is set in the system event status reg.<br>• 01: Assert and Deassert Error pin according to Error Pin Data register<br>• 00: Disable Error pin assertion |
| 3:2 | 0x1<br>RW/P | **ERR1_ENABLE:**<br>• 11: Reserved.<br>• 10: Assert Error Pin when Non-Fatal Error is set in the system event status reg.<br>• 01: Assert and Deassert Error pin according to Error Pin Data register<br>• 00: Disable Error pin assertion |
| 1:0 | 0x1<br>RW/P | **ERR0_ENABLE:**<br>• 11: Reserved.<br>• 10: Assert Error Pin when Correctable Error is set in the system event status reg.<br>• 01: Assert and Deassert Error pin according to Error Pin Data register<br>• 00: Disable Error pin assertion |

## 45.2.25 GLREG Error Pin Status (GLREG_ERRPINSTS)—Offset 258h

This register reflects the state of the error pin assertion. The status bit of the corresponding error pin is set upon the deassertion to assertion transition of the error pin.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 258h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Reserved:** Reserved. |
| 2 | 0x0<br>RW/1C/V/P | **ERR2_STS:** This bit is set upon the transition of deassertion to assertion of the Error pin. Software writes '1' to clear the status. |
| 1 | 0x0<br>RW/1C/V/P | **ERR1_STS:** This bit is set upon the transition of deassertion to assertion of the Error pin. Software writes '1' to clear the status. |
| 0 | 0x0<br>RW/1C/V/P | **ERR0_STS:** This bit is set upon the transition of deassertion to assertion of the Error pin. Software writes '1' to clear the status. |

## 45.2.26 GLREG Error Pin Data (GLREG_ERRPINDATA)—Offset 25Ch

This register provides the data value when the error pin is configured as a general purpose output.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:4, F:0] + 25Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Reserved:** Reserved. |
| 2 | 0x0<br>RW | **ERR2_PINDATA:** (applies when ERRPINCTL[5:4]=01, otherwise reserved)<br>This bit acts as the general purpose output for the ERR[2] pin. ERR [2] pin value will follow the value programmed in ERR[2] Pin Data register.<br>• 0 - Deassert ERR[2] pin<br>• 1 - Assert ERR[2] pin |
| 1 | 0x0<br>RW | **ERR1_PINDATA:** (applies when ERRPINCTL[3:2]=01, otherwise reserved).<br>This bit acts as the general purpose output for the ERR[1] pin. ERR [1] pin value will follow the value programmed in ERR[1] Pin Data register.<br>• 0 - Deassert ERR[1] pin<br>• 1 - Assert ERR[1] pin |
| 0 | 0x0<br>RW | **ERR0_PINDATA:** (applies when ERRPINCNTL[1:0]=01, otherwise reserved).<br>This bit acts as the general purpose output for the ERR[0] pin. Error [0] pin value will follow the value programmed in ERR[0] Pin Data register.<br>• 0 - Deassert ERR[0] pin<br>• 1 - Assert ERR[0] pin |

## 45.2.27 GLREG Uncorrectable Error Counter (GLREG_LUNCERRCNT)—Offset 294h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:4, F:0] + 294h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C/V/P | **ERR_OVF:**<br>• 0: No overflow occurred<br>• 1: Error overflow. The error count may not be valid. |
| 6:0 | 0x0<br>RW/1C/V/P | **ERRCNT:** This counter accumulates errors that occur when the associated error type is selected in the FNBERRCNTSEL register.<br>This field is cleared by writing 7Fh. Maximum counter available is 127d (7Fh).<br>***Note:*** Single bit write clears the corresponding bit. |

§ §

# 46 Root Complex Event Collector - B0, D5, F0

## 46.1 Introduction and Index

The host-accessible registers for the Root Complex Event Collector (RCEC) are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 46.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 5 (decimal), Function 0. The offset addresses are listed.

**Table 46-1. Summary of PCI Configuration Registers—0/5/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "RCEC Vendor ID (RCEC_VID)—Offset 0h" on page 1509 |
| 4 | 2 | 0000 | "RCEC PCI Command (RCEC_PCICMD)—Offset 4h" on page 1510 |
| 6 | 2 | 0010 | "RCEC PCI Status (RCEC_PCISTS)—Offset 6h" on page 1511 |
| 9 | 3 | 00080700 | "RCEC Class Code (RCEC_CC)—Offset 9h" on page 1512 |
| C | 2 | 00 | "RCEC Cacheline Size (RCEC_CLS)—Offset Ch" on page 1512 |
| 3D | 3 | 01 | "RCEC Interrupt Pin (RCEC_INTP)—Offset 3Dh" on page 1513 |
| 40 | 2 | 8010 | "RCEC PCI Express Capability List (RCEC_EXPCAPLST)—Offset 40h" on page 1513 |
| 42 | 2 | 00A2 | "RCEC PCI Express Capability List (RCEC_EXPCAPLST)—Offset 40h" on page 1513 |
| 48 | 2 | 0000 | "RCEC Device Control (RCEC_DEVCTL)—Offset 48h" on page 1515 |
| 4A | 1 | 00 | "RCEC Device Status (RCEC_DEVSTS)—Offset 4Ah" on page 1516 |
| 5C | 2 | 00 | "RCEC Root Control (RCEC_ROOTCTL)—Offset 5Ch" on page 1517 |
| 82 | 2 | C803 | "RCEC Power Management Capabilities (RCEC_PMCAP)—Offset 82h" on page 1518 |
| 84 | 2 | 0000 | "RCEC Power Management Control / Status (RCEC_PMCSR)—Offset 84h" on page 1518 |
| 90 | 2 | 0005 | "RCEC MSI Capability List (RCEC_MSICAPLST)—Offset 90h" on page 1519 |
| 92 | 2 | 0100 | "RCEC MSI Message Control (RCEC_MSICTL)—Offset 92h" on page 1519 |
| 94 | 4 | 00000000 | "RCEC MSI Message Address (RCEC_MSIADDR)—Offset 94h" on page 1520 |
| 98 | 2 | 0000 | "RCEC MSI Message Data (RCEC_MSIDATA)—Offset 98h" on page 1520 |
| 9C | 4 | 00000000 | "RCEC MSI Mask Bit (RCEC_MSIMSK)—Offset 9Ch" on page 1521 |
| A0 | 2 | 0000 | "RCEC MSI Pending Bit (RCEC_MSIPENDING)—Offset A0h" on page 1521 |
| 100 | 4 | 15010001 | "RCEC Advanced Error Reporting Extended Capability Header (RCEC_AERCAPHDR)—Offset 100h" on page 1521 |
| 104 | 4 | 00000000 | "RCEC Uncorrectable Error Status (RCEC_ERRUNCSTS)—Offset 104h" on page 1522 |
| 110 | 2 | 0000 | "RCEC Correctable Error Status (RCEC_ERRCORSTS)—Offset 110h" on page 1523 |
| 114 | 2 | E000 | "RCEC Correctable Error Mask (RCEC_ERRCORMSK)—Offset 114h" on page 1524 |
| 118 | 2 | 0000 | "RCEC Advanced Error Capabilities and Control (RCEC_AERCAPCTL)—Offset 118h" on page 1525 |
| 11C | 4 | 00000000 | "RCEC Header Log (RCEC_AERHDRLOG1)—Offset 11Ch" on page 1525 |
| 120 | 4 | 00000000 | "RCEC Header Log (RCEC_AERHDRLOG2)—Offset 120h" on page 1526 |
| 124 | 4 | 00000000 | "RCEC Header Log (RCEC_AERHDRLOG3)—Offset 124h" on page 1526 |
| 128 | 4 | 00000000 | "RCEC Header Log (RCEC_AERHDRLOG4)—Offset 128h" on page 1526 |
| 12C | 1 | 00 | "RCEC Root Error Command (RCEC_ROOTERRCMD)—Offset 12Ch" on page 1527 |
| 130 | 4 | 00000000 | "RCEC Root Error Status (RCEC_ROOTERRSTS)—Offset 130h" on page 1528 |
| 134 | 4 | 00000000 | "RCEC Error Source Identification (RCEC_ERRSRCID)—Offset 134h" on page 1529 |

## 46.2 Registers in Configuration Space

### 46.2.1 RCEC Vendor ID (RCEC_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **VID:** This field identifies Intel as the manufacturer of the device. |

## 46.2.2 RCEC PCI Command (RCEC_PCICMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved:** Reserved. |
| 10 | 0x0<br>RW | **INTxD:** This bit controls the ability of the PCI-Express Function to generate INTx interrupt message. When set, functions are prevented from asserting INTx interrupt messages.<br>Any INTx emulation interrupts already asserted by the function must be deserted when this bit is set by generating a Deassert_INTx message(s). |
| 9 | 0x0<br>RO | **FBE:** Not applicable to PCI-Express. Hardwired to 0. |
| 8 | 0x0<br>RW | **SEE:** (SERR Enable) When set, this bit enables reporting of Non-Fatal and Fatal Errors detected by the Function of the Root Complex. ERR_COR messages are not affected by this bit.<br>Errors are reported when enabled either through this bit or through the PCI Express-specific bits in the Device Control Register (RCEC_DEVCTL). |
| 7 | 0x0<br>RO | **WCC:** Not applicable to PCI-Express. Hardwired to 0. |
| 6 | 0x0<br>RW | **PERE:** This bit controls the setting of the master data parity error bit in the Status Register (RCEC_PCISTS) in response to a parity error received on the PCI Express interface (poisoned TLP). |
| 5 | 0x0<br>RO | **VGA_PSE:** Not applicable to PCI-Express. Hardwired to 0. |
| 4 | 0x0<br>RO | **MWIE:** Not applicable to PCI-Express. Hardwired to 0. |
| 3 | 0x0<br>RO | **SCE:** Not applicable to PCI-Express. Hardwired to 0. |
| 2 | 0x0<br>RW | **BME:** This bit controls the ability of the Function to issue Memory and I/O read or write requests, and the ability of Root or Switch port to forward memory and I/O read or write requests in the upstream direction. MSI interrupts are inband memory writes and are blocked when this bit is 0b. |
| 1 | 0x0<br>RO | **MSE:** This bit controls the function's response to Memory Space accesses. When this bit is 0b, the function will handle memory transactions targeting the Function as an Unsupported Request (UR). |
| 0 | 0x0<br>RO | **IOSE:** This bit controls the function's response to IO Space accesses. When this bit is 0b, the function will handle memory transactions targeting the Function as an Unsupported Request (UR). |

## 46.2.3 RCEC PCI Status (RCEC_PCISTS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **DPE:** This bit is set when a poisoned TLP is received. This bit is set even when the parity error response enable, bit 6 of the PCICMD Register (RCEC_PCICMD) is not set. |
| 14 | 0x0<br>RW/1C/V | **SSE:** This bit is set when ERR_FATAL or ERR_NONFATAL messages are sent to the root complex event collector and the SERR enable bit in the PCICMD Register (RCEC_PCICMD) is set. |
| 13 | 0x0<br>RO | **RMA:** This bit is set when the requester receives a completion with an UR completion status. |
| 12 | 0x0<br>RO | **RTA:** This bit is set when a requester receives a CA completions status. |
| 11 | 0x0<br>RO | **STA:** This bit is set when the switch generates a completion packet with Completer Abort (CA) status is generated by its primary side. |
| 10:9 | 0x0<br>RO | **DVT:** These bits have no meaning on PCI Express. Fast decode timing is reported. |
| 8 | 0x0<br>RO | **MDPD:** This bit is set by a requester (primary side for type1 configuration header functions) if the parity error response enable bit (PERE) in the Command Register (RCEC_PCICMD) is set and either of the following two conditions occur:<br>• Requester receives a completion marked poisoned.<br>• Requester poisons a write requests.<br>If the parity error bit is 0b, this bit is never set. |
| 7 | 0x0<br>RO | **FBC:** This bit has no meaning on PCI Express. |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **C66:** This bit has no meaning on PCI Express. |
| 4 | 0x1<br>RO | **CAPE:** This bit indicates the presence of an Extended capabilities list items. |
| 3 | 0x0<br>RO/V | **INTS:** When set, this bit indicates that an INTx emulation interrupt is pending internally in this function. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 46.2.4 RCEC Class Code (RCEC_CC)—Offset 9h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 9h

**Default:** 00080700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x8<br>RW/O | **BC:** Root Complex Event Collector |
| 15:8 | 0x7<br>RW/O | **SC:** Root Complex Event Collector |
| 7:0 | 0x0<br>RW/O | **RLPI:** Reserved. |

## 46.2.5 RCEC Cacheline Size (RCEC_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:5, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **CLS:** These bits specify the system cache-line size in units of DWords.<br>This field is implemented by PCI Express devices but has no effect on device behavior. |

## 46.2.6 RCEC Interrupt Pin (RCEC_INTP)—Offset 3Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:5, F:0] + 3Dh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RO | **INTP:** This register tells which interrupt pin the function uses.<br>• 01h: Generate INTA<br>• 02h: Generate INTB<br>• 03h: Generate INTC<br>• 04h: Generate INTD<br>• Others: Reserved<br>Devices (or device functions) that do not use an interrupt pin must put a 0 in this register. As a single function device, this device specifies INTA as its interrupt pin. |

## 46.2.7 RCEC PCI Express Capability List (RCEC_EXPCAPLST)—Offset 40h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 40h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x80<br>RO | **NP:** Contains the offset of the next item in the capabilities list (RCEC_PMCAPLST). |
| 7:0 | 0x10<br>RO | **CAPID:** Identifies the function as PCI Express capable. |

## 46.2.8 RCEC PCI Express Capabilities (RCEC_EXPCAP)—Offset 42h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 42h

**Default:** 00A2h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13:9 | 0x0<br>RO | **IMN:** This field indicates the interrupt message number that is generated from the PCI Express port. |
| 8 | 0x0<br>RO | **SI:** Indicates the PCI Express link associated with this port is connected to a slot. |
| 7:4 | 0xa<br>RO | **DT:** 1010b Root Complex Event Collector |
| 3:0 | 0x2<br>RO | **VN:** These bits indicate the version number of the PCI Express capability structure. |

## 46.2.9 RCEC Device Control (RCEC_DEVCTL)—Offset 48h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 48h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RO | **MRRS:** This field sets the maximum Read Requests size for the function as a requester. The Function must not generate read requests with size exceeding the set value.<br>• 000b: 128 bytes maximum Read Request size<br>• 001b: 256 bytes maximum Read Request size<br>• 010b: 512 bytes maximum Read Request size<br>• 011b: 1024 bytes maximum Read Request size<br>• 100b: 2048 bytes maximum Read Request size<br>• 101b: 4096 bytes maximum Read Request size<br>• Others: Reserved<br>Functions that do not generate Read Requests larger than 128B and functions that do not generate Read Requests on their own behalf are permitted to implement this field as Read Only (RO) with a value of 000b. |
| 11 | 0x0<br>RO | **Reserved:** Reserved |
| 10 | 0x0<br>RO | **AUXPME:** Not supported |
| 9 | 0x0<br>RO | **Reserved:** Reserved |
| 8 | 0x0<br>RO | **ETFE:** When set, this bit enables a function to use an 8-bit tag field as a Requester. |
| 7:5 | 0x0<br>RW | **MPS:** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as larger as the set value. As a Transmitter, the function must not generate TLPs exceeding the set value.<br>• 000b: 128 bytes maximum payload size<br>• 001b: 256 bytes maximum payload size<br>• 010b: 512 bytes maximum payload size (Unsupported)<br>• 011b: 1024 bytes maximum payload size (Unsupported)<br>• 100b: 2048 bytes maximum payload size (Unsupported)<br>• 101b: 4096 bytes maximum payload size (Unsupported)<br>• Others: Reserved |
| 4 | 0x0<br>RO | **ENRO:** When set, the function is permitted to set the relaxed ordering bit in the attribute field of transactions it initiates that do not require strong write ordering. |
| 3 | 0x0<br>RW | **URRE:** This bit controls the enabling of ERR_CORR, ERR_NONFATAL or ERR_FATAL messages on PCI Express for reporting Unsupported Request (UR) errors. |
| 2 | 0x0<br>RW | **FERE:** When this bit is set, generation of the ERR_FATAL message is enabled. |
| 1 | 0x0<br>RW | **NFERE:** When this bit is set, generation of the ERR_NONFATAL message is enabled. |
| 0 | 0x0<br>RW | **CERE:** When this bit is set, generation of the ERR_CORR message is enabled. |

## 46.2.10 RCEC Device Status (RCEC_DEVSTS)—Offset 4Ah

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:5, F:0] + 4Ah

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **Reserved:** Reserved |
| 4 | 0x0<br>RO | **APD:** Auxiliary Power is not supported. |
| 3 | 0x0<br>RW/1C/V | **URD:** This bit indicates that the device received an Unsupported Request (UR).<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control Register. |
| 2 | 0x0<br>RW/1C/V | **FED:** This bit indicates status of Fatal Errors detected.<br>Errors are logged in this register regardless of the settings of the Correctable Error mask register. |
| 1 | 0x0<br>RW/1C/V | **NFED:** This bit indicates status of Non-Fatal Errors detected. |
| 0 | 0x0<br>RW/1C/V | **CED:** This bit indicates status of Correctable Errors detected. |

## 46.2.11 RCEC Root Control (RCEC_ROOTCTL)—Offset 5Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:5, F:0] + 5Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **RSVD0:** Reserved |
| 4 | 0x0<br>RO | **CRSSVE:** This bit, when set, enables the Root Port to return Configuration Retry Status (CRS) Completion status to software.<br>• 1: Enable software to received a CRS status. This allows software to make the decision to re-issue the configuration request or move on and re-issue the request at a later time.<br>• 0: Disable software from receiving a CRS response. The PCI-E will wait and re-issue the configuration request until it receives a response other than CRS without notifying software. |
| 3 | 0x0<br>RO | **PMEIE:** This field controls the generation of interrupts for PME messages.<br>• 1: Enables interrupt generation upon receipt of a PME message as reflected in the PME Status bit defined in the ROOTSTS register. A PME interrupt is generated if the PMESTATUS register bit is set when this bit is set from a cleared state.<br>• 0: Disables interrupt generation for PME messages. |
| 2 | 0x0<br>RW | **SEFEE:** This field controls generation of system errors in the PCI-Express root Complex Event Collector for Fatal Errors.<br>• 1: indicates that a System Error should be generated if a Fatal Error (ERR_FATAL) is reported by any of the devices associated with and including this PCI-Express root complex event collector.<br>• 0: No System Error should be generated on a Fatal Error (ERR_FATAL) reported by any of the devices associated with and including this PCI-Express root complex event collector. |
| 1 | 0x0<br>RW | **SENFEE:** This field controls generation of system errors in the PCI-Express root Complex Event Collector for Non-Fatal Errors.<br>• 1: indicates that a System Error should be generated if a Non-Fatal Error (ERR_NONFATAL) is reported by any of the devices associated with and including this PCI-Express root complex event collector.<br>• 0: No System Error should be generated on a Non-Fatal Error (ERR_NONFATAL) reported by any of the devices associated with and including this PCI-Express root complex event collector. |
| 0 | 0x0<br>RW | **SECEE:** This field controls generation of system errors in the PCI-Express root Complex Event Collector for Correctable Errors.<br>• 1: indicates that a System Error should be generated if a Correctable Error (ERR_COR) is reported by any of the devices associated with and including this PCI-Express root complex event collector<br>• 0: No System Error should be generated on a Correctable Error (ERR_COR) reported by any of the devices associated with and including this PCI-Express root complex event collector. |

## 46.2.12 RCEC Power Management Capabilities (RCEC_PMCAP)—Offset 82h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 82h

**Default:** C803h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x19<br>RO | **PMESUP:** PME assertion is supported when in D3hot.<br>PME assertion from D3cold is not supported. |
| 10 | 0x0<br>RO | **D2S:** Not supported |
| 9 | 0x0<br>RO | **D1S:** Not supported |
| 8:6 | 0x0<br>RO | **AC:** Auxiliary power is not supported. |
| 5 | 0x0<br>RO | **DSI:** Device-specific initialization is not required when transitioning to D0 from D3hot state.<br>This bit is zero. |
| 4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x0<br>RO | **PMECLK:** Does not apply to PCI Express.<br>Hard-wired to 0. |
| 2:0 | 0x3<br>RO | **VER:** PM implementation is compliant with PCI Bus Power Management Interface Specification, Revision 11 1.2. |

## 46.2.13 RCEC Power Management Control / Status (RCEC_PMCSR)—Offset 84h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 84h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V/P | **PMESTS:** Reserved. |
| 14:13 | 0x0<br>RO | **DC:** Not supported |
| 12:9 | 0x0<br>RO | **DS:** Not supported |
| 8 | 0x0<br>RW/P | **PMEEN:** Gates assertion of the PME message. |
| 7:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1:0 | 0x0<br>RW/V | **PS:** This field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the supported values is given below:<br>• 0h - D0<br>• 3h - D3hot |

## 46.2.14 RCEC MSI Capability List (RCEC_MSICAPLST)—Offset 90h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 90h

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **NP:** Contains the offset of the next item in the capabilities list. A null value is used to indicate that this is the last capability. |
| 7:0 | 0x5<br>RO | **CAPID:** Identifies the function as MSI capable. |

## 46.2.15 RCEC MSI Message Control (RCEC_MSICTL)—Offset 92h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 92h

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x1<br>RO | **PVM:** This bit indicates that this device supports MSI per-vector masking. |
| 7 | 0x0<br>RO | **AD64C:** When set, this bit indicates that the function is capable of generating a 64-bit message address. |
| 6:4 | 0x0<br>RW | **MMEN:** Only one message is supported.<br>These bits are R/W for software compatibility. |
| 3:1 | 0x0<br>RO | **MMC:** Only one message is supported. |
| 0 | 0x0<br>RW | **MSIE:** When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. |

## 46.2.16 RCEC MSI Message Address (RCEC_MSIADDR)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address:** Message address specified by the system, always DWORD aligned |
| 1:0 | 0x0<br>RO | **Reserved:** Reserved. |

## 46.2.17 RCEC MSI Message Data (RCEC_MSIDATA)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **TM:**<br>• 0: Edge Triggered<br>• 1: Level Triggered |
| 14 | 0x0<br>RW | **LVL:** if TM is 0h, then this field is a don't care.<br>Edge triggered messages are always treated as assert messages. For level triggered interrupts, this bit reflects the state of the interrupt input if TM is 1h, then:<br>• 0: Deassert Messages<br>• 1: Assert Messages |
| 13:12 | 0x0<br>RW | **Reserved:** Reserved |
| 11:8 | 0x0<br>RW | **DM:**<br>• 0000: Fixed: Trigger Mode can be edge or level.<br>• 0001: Lowest Priority: Trigger Mode can be edge or level.<br>• 0010: SMI/PMI/MCA - Not supported via MSI of root port<br>• 0011: Reserved - Not supported via MSI of root port<br>• 0100: NMI - Not supported via MSI of root port<br>• 0101: INIT - Not supported via MSI of root port<br>• 0110: Reserved<br>• 0111: ExtINT - Not supported via MSI of primary port<br>• 1000 -1111: Reserved |
| 7:0 | 0x0<br>RW | **Reserved:** Reserved |

## 46.2.18 RCEC MSI Mask Bit (RCEC_MSIMSK)—Offset 9Ch

The MSI Mask Bit register enables software to disable message sending on a per-vector basis.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RW | **MSKB:** For each Mask bit that is set, the device is prohibited from sending the associated message. Corresponding bits are masked if set to '1' |

## 46.2.19 RCEC MSI Pending Bit (RCEC_MSIPENDING)—Offset A0h

The Mask Pending register enables software to defer message sending on a per-vector basis.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + A0h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RO/V | **PB:** For each Pending bit that is set, the device has a pending associated message. Corresponding bits are pending if set to '1' |

## 46.2.20 RCEC Advanced Error Reporting Extended Capability Header (RCEC_AERCAPHDR)—Offset 100h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 100h

**Default:** 15010001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x150<br>RO | **NCO:** Contains the offset of the next structure in the Extended Capabilities list (RCEC_RCECEPACAPHDR). |
| 19:16 | 0x1<br>RO | **CV:** Indicates the version of the Capability structure present. |
| 15:0 | 0x1<br>RO | **EXCAPID:** Identifies the function as Advanced Error Reporting (AER) capable. |

## 46.2.21 RCEC Uncorrectable Error Status (RCEC_ERRUNCSTS)—Offset 104h

This register reports the error status of individual Uncorrectable Error sources. An individual error status bit that is set to 1 indicates that a particular error occurred. Software can clear an error status by writing a 1 to the respective bit.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x0<br>RW/1C/V/P | **UIE:** This bit is set whenever an IEH Uncorrectable Internal Error as defined in IEH_UNCERRSTS is detected. |
| 21 | 0x0<br>RO | **RSVD1:** Reserved |
| 20 | 0x0<br>RW/1C/V/P | **URE:** As a receiver, Set whenever an Unsupported Request (negatively-decoded transaction) is detected. The Header is logged. |
| 19 | 0x0<br>RO | **RSVD2:** Reserved |
| 18 | 0x0<br>RW/1C/V/P | **MalformedTLP:** As a receiver, set whenever a malformed TLP (Length not equal to 000000001b for config requests) is detected. The Header is logged. |
| 17 | 0x0<br>RO | **RSVD3:** Reserved |
| 16 | 0x0<br>RW/1C/V/P | **UnexpectedCompletion:** As a receiver, set whenever a completion is received that does not match the requestor ID or outstanding Tag. The Header is logged. |
| 15:0 | 0x0<br>RO | **RSVD4:** Reserved |

## 46.2.22 RCEC Correctable Error Status (RCEC_ERRCORSTS)—Offset 110h

This register reports the error status of individual Correctable Error sources on a PCI Express device. An individual error status bit set to 1 indicates that a particular error has occurred. Software can clear the error status by writing a 1 to the respective bit.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 110h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **HLOE:** Reserved. |
| 14 | 0x0<br>RO | **CIE:** Reserved. |
| 13 | 0x0<br>RW/1C/V/P | **ANFE:** Reserved. |
| 12 | 0x0<br>RO | **RTTE:** The PCI-E sets this bit when replay timer time-out occurs. |
| 11:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RO | **RNRE:** The PCI-E sets this bit when the replay number rolls over from 11 to 00. |
| 7 | 0x0<br>RO | **BDLLPE:** The switch sets this bit on CRC errors on DLLP. |
| 6 | 0x0<br>RO | **BTLPE:** The switch sets this bit on CRC errors on TLP. |
| 5:1 | 0x0<br>RO | **RSVD1:** Reserved |
| 0 | 0x0<br>RO | **RE:** The PCI-E sets this bit when the physical layer detects a receiver error. |

## 46.2.23 RCEC Correctable Error Mask (RCEC_ERRCORMSK)—Offset 114h

This register controls the reporting of individual Correctable Errors via ERR_COR message. A masked error (respective bit set in mask register) is not reported to the host bridge by the switch. There is a mask bit corresponding to every bit in the Correctable Error Status Register.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 114h

**Default:** E000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x1<br>RO | **HLOEM:** Reserved. |
| 14 | 0x1<br>RO | **CIEM:** Reserved. |
| 13 | 0x1<br>RW/P | **ANFEM:** This bit is set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting. |
| 12 | 0x0<br>RO | **RTTEM:** Reserved. |
| 11:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RO | **RNREM:** Reserved. |
| 7 | 0x0<br>RO | **BDLLPEM:** Reserved. |
| 6 | 0x0<br>RO | **BTLPEM:** Reserved. |
| 5:1 | 0x0<br>RO | **RSVD1:** Reserved |
| 0 | 0x0<br>RO | **REM:** Reserved. |

## 46.2.24 RCEC Advanced Error Capabilities and Control (RCEC_AERCAPCTL)—Offset 118h

This register gives the status and control for ECRC checks and also the pointer to the first Uncorrectable Error that happened.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:5, F:0] + 118h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **MHRE:** When set, this bit enables the function to record more than one error header. |
| 9 | 0x0<br>RO | **MHRC:** Supports more than one error header. |
| 8 | 0x0<br>RO | **ECE:** Not supported. Hard-wired to 0 |
| 7 | 0x0<br>RO | **ECC:** Not supported. Hard-wired to 0 |
| 6 | 0x0<br>RO | **EGE:** Not supported. Hard-wired to 0 |
| 5 | 0x0<br>RO | **EGC:** Not supported. Hard-wired to 0 |
| 4:0 | 0x0<br>RO/V/P | **FEP:** This field identifies the bit position of the first error reported in the Uncorrectable Error Status Register.<br>This register re-arms itself (which does not change its current value) as soon as the error status bit indicated by the pointer is cleared by the software by writing a 1 to that status bit. |

## 46.2.25 RCEC Header Log (RCEC_AERHDRLOG1)—Offset 11Ch

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 11Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLPHDRLOG:** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 46.2.26 RCEC Header Log (RCEC_AERHDRLOG2)—Offset 120h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLPHDRLOG:** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 46.2.27 RCEC Header Log (RCEC_AERHDRLOG3)—Offset 124h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLPHDRLOG:** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 46.2.28 RCEC Header Log (RCEC_AERHDRLOG4)—Offset 128h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLPHDRLOG:** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 46.2.29 RCEC Root Error Command (RCEC_ROOTERRCMD)—Offset 12Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:5, F:0] + 12Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **FERE:** When set, this bit enables the generation of an interrupt when a Fatal Error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 1 | 0x0<br>RW | **NFERE:** When set, this bit enables the generation of an interrupt when a Non-Fatal Error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 0 | 0x0<br>RW | **CERE:** When set, this bit enables the generation of an interrupt when a Correctable Error is reported by any of the functions in the hierarchy associated with this Root Port. |

## 46.2.30 RCEC Root Error Status (RCEC_ROOTERRSTS)—Offset 130h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **AEMN:** This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability.<br><br>For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register.<br><br>For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant.<br><br>• If both MSI and MSI-X are implemented, they are permitted to use different vectors, though software is permitted to enable only one mechanism at a time. If software enables both MSI and MSI-X at the same time, the value in this register is undefined.<br>• If MSI-X is enabled, the value in this register must indicate the vector for MSI-X.<br>• If MSI is enabled or neither is enabled, the value in this register must indicate the vector for MSI. |
| 26:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW/1C/V/P | **FEMR:** Set when one or more Fatal Uncorrectable Error Messages have been received. |
| 5 | 0x0<br>RW/1C/V/P | **NFEMR:** Set when one or more Non-Fatal Uncorrectable Error Messages have been received. |
| 4 | 0x0<br>RW/1C/V/P | **FUF:** Set when the first Uncorrectable Error Message or PCI_error Message received is for a Fatal Error. |
| 3 | 0x0<br>RW/1C/V/P | **MEFR:** Set when either a Fatal or a Non-Fatal Error is received and ERR_FATAL/NONFATAL Received is already Set. |
| 2 | 0x0<br>RW/1C/V/P | **EFR:** Set when a Fatal or a Non-Fatal Error Message is received and this bit is not already Set. |
| 1 | 0x0<br>RW/1C/V/P | **MCER:** Set when a Correctable Error Message is received and ERR_COR Received is already Set. |
| 0 | 0x0<br>RW/1C/V/P | **CER:** Set when a Correctable Error Message is received and this bit is not already Set. |

### 46.2.31 RCEC Error Source Identification (RCEC_ERRSRCID)—Offset 134h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:5, F:0] + 134h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO/V/P | **EFSID:** Requester ID of the source when a Fatal or Non-Fatal Error is received and the Fatal Error Message Received (FEMR) or Non-Fatal Error Message Received (NFEMR) bit is not already set. |
| 15:0 | 0x0<br>RO/V/P | **ECSID:** Requester ID of the source when a Correctable Error is received and the Correctable Error Received (CER) bit is not already set. |

§ §

# 47 RP - Intel® QuickAssist Technology - B0, D6/D22/D23, F0

## 47.1 Introduction and Index

The host-accessible registers for the "Virtual Root Port" (vRP) dedicated to the Intel® QuickAssist Technology Controller are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The SoC also contains a vRP (Device 22) dedicated to LAN Controller 0 and another vRP (Device 23) dedicated to LAN Controller 1.

The registers shown in this chapter apply to vRP Device 6 as well as vRP Devices 22 and 23. Register information unique to a device is explained in the register and bit field descriptions.

The legend for the register access codes are described in Chapter 43, "Introduction."

### 47.1.1 Host Configuration Space

These Virtual Root Port (vRP) registers are discovered in Host Configuration Space starting at:

- Bus 0, Device 6, Function 0 for the vRP of the Intel® QuickAssist Technology Controller
- Bus 0, Device 22 (decimal), Function 0 for the vRP of LAN Controller 0
- Bus 0, Device 23 (decimal), Function 0 for the vRP of LAN Controller 1

**Table 47-1. Summary of PCI Configuration Registers—0/6/0, 0/22/0, and 0/23/0 (Sheet 1 of 3)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "VID—Offset 0h" on page 1534 |
| 2 | 2 | See register | "DID—Offset 2h" on page 1534 |
| 4 | 2 | 0000 | "PCICMD—Offset 4h" on page 1535 |
| 6 | 2 | 0010 | "PCISTS—Offset 6h" on page 1536 |
| 9 | 3 | 00060400 | "CCR—Offset 9h" on page 1537 |
| C | 1 | 00 | "CLS—Offset Ch" on page 1537 |
| 1C | 1 | 00 | "IOBASE—Offset 1Ch" on page 1538 |
| 1D | 1 | 00 | "IOLIMIT—Offset 1Dh" on page 1538 |
| 20 | 2 | 0000 | "MEMBASE—Offset 20h" on page 1539 |
| 22 | 2 | 0000 | "MEMLIMIT—Offset 22h" on page 1540 |
| 24 | 2 | 0001 | "PFBASE—Offset 24h" on page 1541 |
| 26 | 2 | 0001 | "PFLIMIT—Offset 26h" on page 1542 |
| 28 | 4 | 00000000 | "PFBASEU—Offset 28h" on page 1542 |
| 2C | 4 | 00000000 | "PFLIMITU—Offset 2Ch" on page 1543 |
| 34 | 1 | 40 | "CAPPTR—Offset 34h" on page 1543 |
| 3C | 1 | 00 | "INTL—Offset 3Ch" on page 1544 |
| 3D | 1 | 01 | "INTP—Offset 3Dh" on page 1544 |
| 3E | 2 | 0000 | "BCTL—Offset 3Eh" on page 1544 |
| 40 | 2 | 8010 | "EXPCAPLST—Offset 40h" on page 1546 |
| 42 | 2 | 0042 | "EXPCAP—Offset 42h" on page 1547 |
| 44 | 4 | 00008001 | "DEVCAP—Offset 44h" on page 1548 |
| 48 | 2 | 2000 | "DEVCTL—Offset 48h" on page 1549 |
| 4A | 2 | 0000 | "DEVSTS—Offset 4Ah" on page 1550 |
| 4C | 4 | See register | "LINKCAP—Offset 4Ch" on page 1551 |
| 50 | 2 | 0000 | "LINKCTL—Offset 50h" on page 1553 |
| 52 | 2 | 1011 | "LINKSTS—Offset 52h" on page 1554 |
| 5A | 2 | 0040 | "SLOTSTS—Offset 5Ah" on page 1555 |
| 5C | 2 | 0000 | "ROOTCTL—Offset 5Ch" on page 1556 |
| 5E | 2 | 0001 | "ROOTCAP—Offset 5Eh" on page 1557 |
| 60 | 4 | 00000000 | "ROOTSTS—Offset 60h" on page 1557 |
| 64 | 4 | 00000020 | "DEVCAP2—Offset 64h" on page 1558 |
| 68 | 2 | 0000 | "DEVCTL2—Offset 68h" on page 1560 |

## Table 47-1. Summary of PCI Configuration Registers—0/6/0, 0/22/0, and 0/23/0 (Sheet 2 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 6C | 4 | 00000002 | "LINKCAP2—Offset 6Ch" on page 1561 |
| 70 | 2 | 0001 | "LINKCTL2—Offset 70h" on page 1562 |
| 72 | 2 | 0000 | "LINKSTS2—Offset 72h" on page 1564 |
| 80 | 2 | 8801 | "PMCAPLST—Offset 80h" on page 1564 |
| 82 | 2 | C803 | "PMCAPLST—Offset 80h" on page 1564 |
| 84 | 2 | 0008 | "PMCSR—Offset 84h" on page 1566 |
| 86 | 2 | 00 | "PMBSE—Offset 86h" on page 1566 |
| 88 | 2 | 900D | "SSCAPLST—Offset 88h" on page 1567 |
| 8C | 2 | 8086 | "SSVID—Offset 8Ch" on page 1568 |
| 8E | 2 | 0000 | "SSID—Offset 8Eh" on page 1568 |
| 90 | 2 | 0005 | "MSICAPLST—Offset 90h" on page 1569 |
| 92 | 2 | 0100 | "MSICTL—Offset 92h" on page 1569 |
| 94 | 4 | 00000000 | "MSIADDR—Offset 94h" on page 1570 |
| 98 | 4 | 00000000 | "MSIDATA—Offset 98h" on page 1570 |
| 9C | 4 | 00000000 | "MSIMSK—Offset 9Ch" on page 1571 |
| A0 | 4 | 00000000 | "MSIPENDING—Offset A0h" on page 1571 |
| A8 | 4 | 00000000 | "RPPMCSR—Offset A8h" on page 1572 |
| B0 | 2 | 0000 | "PMCSRS—Offset B0h" on page 1572 |
| C8 | 4 | 00000000 | "SMICSR—Offset C8h" on page 1573 |
| CC | 4 | 00000000 | "SCICSR—Offset CCh" on page 1573 |
| D8 | 4 | 00000000 | "PBTXNCTL—Offset D8h" on page 1574 |
| EA | 2 | 0000 | "PLKCTL—Offset EAh" on page 1574 |
| EC | 4 | 00000000 | "LTRCSR—Offset ECh" on page 1575 |
| F0 | 4 | 00000000 | "LTRL—Offset F0h" on page 1576 |
| F4 | 4 | 00000000 | "LTROVR—Offset F4h" on page 1577 |
| F8 | 1 | 00 | "INTXSWZCTL—Offset F8h" on page 1578 |
| 118 | 4 | 00000000 | "AERCAPCTL—Offset 118h" on page 1579 |
| 11C | 4 | 00000000 | "AERHDRLOG1—Offset 11Ch" on page 1579 |
| 120 | 4 | 00000000 | "AERHDRLOG2—Offset 120h" on page 1580 |
| 124 | 4 | 00000000 | "AERHDRLOG3—Offset 124h" on page 1580 |
| 128 | 4 | 00000000 | "AERHDRLOG4—Offset 128h" on page 1580 |
| 12C | 4 | 00000000 | "ROOTERRCMD—Offset 12Ch" on page 1581 |
| 130 | 4 | 00000000 | "ROOTERRSTS—Offset 130h" on page 1581 |
| 134 | 4 | 00000000 | "ERRSRCID—Offset 134h" on page 1582 |
| 138 | 4 | 0001000D | "ACSCAPHDR—Offset 138h" on page 1582 |
| 13C | 2 | 005F | "ACSCAPHDR—Offset 138h" on page 1582 |
| 13E | 2 | 0000 | "ACSCTL—Offset 13Eh" on page 1584 |
| 2DC | 4 | 00008000 | "XPTDEF—Offset 2DCh" on page 1585 |
| 2E0 | 4 | 00001D00 | "XPTDEF2—Offset 2E0h" on page 1587 |

## Table 47-1. Summary of PCI Configuration Registers—0/6/0, 0/22/0, and 0/23/0 (Sheet 3 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2E4 | 2 | 0010 | "IOSFDEVCLKGCTL—Offset 2E4h" on page 1589 |
| 2E6 | 2 | 8010 | "SBDEVCLKGCTL—Offset 2E6h" on page 1589 |
| 2E8 | 2 | 0010 | "IDFDEVCLKGCTL—Offset 2E8h" on page 1590 |

## 47.2 Registers in Configuration Space

### 47.2.1 VID—Offset 0h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 0h
**Offset:** [B:0, D:22, F:0] + 0h
**Offset:** [B:0, D:23, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies Intel as the manufacturer of the device. |

### 47.2.2 DID—Offset 2h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 2h
**Offset:** [B:0, D:22, F:0] + 2h
**Offset:** [B:0, D:23, F:0] + 2h

**Default:** 19A3h for D6
**Default:** 19D1h for D22
**Default:** 19D2h for D23

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | D6: 0x19A3<br>D22: 0x19D1<br>D23: 0x19D2<br>RO/V | **Device ID (DID):** This field identifies the particular function as allocated by Intel. |

## 47.2.3 PCICMD—Offset 4h

This register controls how the device behaves on the primary interface (PCI Express).

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 4h
**Offset:** [B:0, D:22, F:0] + 4h
**Offset:** [B:0, D:23, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (INTxD):** This bit controls the ability of the PCI-Express Function to generate legacy INTx interrupt message.<br>When set, functions are prevented from asserting INTx interrupt messages. Any INTx emulation interrupts already asserted by the function must be deasserted when this bit is set by generating a Deassert_INTx message(s). |
| 9 | 0x0<br>RO | **Fast Back-to-back enable (FBE):** Not applicable to PCI-Express. Hardwired to 0. |
| 8 | 0x0<br>RW | **SERR# Enable (SEE):** When set, this bit enables reporting of Non-Fatal and Fatal errors detected by the Function of the Root Complex. |
| 7 | 0x0<br>RO | **Wait Cycle Control (WCC):** Not applicable to PCI-Express. Hardwired to 0. |
| 6 | 0x0<br>RW | **Parity Error Response Enable (PERE):** This bit controls the setting of the master data parity error bit in the Status Register PCI Status Register (PCISTS) in response to a parity error received on the PCI Express interface (either internal queue errors or a poisoned TLP). |
| 5 | 0x0<br>RO | **VGA Palette Snoop Enable (VGA_PSE):** Not applicable to PCI-Express. Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not applicable to PCI-Express. Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not applicable to PCI-Express. Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** This bit controls the ability of the Function to issue Memory and I/O read or write requests, and the ability of Root or Switch port to forward memory and I/O read or write requests in the upstream direction.<br>When this bit is 0b, memory and I/O requests received at the root port or downstream side of a switch port (secondary side) must be handled as an Unsupported Request (UR).<br>MSI interrupts are inband memory writes and are blocked when this bit is 0b. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):**<br>• 0 - Respond to all Memory Requests on the primary interface as Unsupported Request Received. Forward all memory requests from the secondary interface to the primary interface.<br>• 1- Enable forwarding of memory transactions to the secondary interface and any internal function. |
| 0 | 0x0<br>RW | **I/O Space Enable (IOSE):**<br>• 0 - Respond to all I/O Requests on the primary interface with an Unsupported Request Completion. Forward all I/O transactions from the secondary interface to the primary interface (if supported).<br>• 1 - Enable forwarding of I/O Requests to the secondary interface. |

## 47.2.4 PCISTS—Offset 6h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 6h
**Offset:** [B:0, D:22, F:0] + 6h
**Offset:** [B:0, D:23, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Detected Parity Error (DPE):** This bit is set when a poisoned TLP is received from PCIe or IOSF.<br>This bit is set even when the parity error response enable bit (bit[6] of the PCI Command Register (PCICMD) is not set. |
| 14 | 0x0<br>RW/1C/V | **Signaled System Error (SSE):** This bit is set when ERR_FATAL or ERR_NONFATAL messages are sent to the root complex and the SERR enable bit in the PCICMD Register PCI Command Register (PCICMD) is set. |
| 13 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit is set when the requester receives a completion with an UR completion status. |
| 12 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit is set when a requester receives a CA completions status. |
| 11 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit is Set when the port completes a Posted or Non-Posted Request as a Completer Abort error. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DVT):** These bits have no meaning on PCI Express. |
| 8 | 0x0<br>RW/1C/V | **Master Data Parity Error Detected (MDPD):** This bit is set if the parity error response enable bit (PERE) in the Command Register PCI Command Register (PCICMD) is set and either of the following two conditions occur:<br>• Requester receives a completion marked poisoned.<br>• Requester sends a poisoned request (includes writes and messages).<br>If the parity error bit is 0b, this bit is never set. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** This bit has no meaning on PCI Express. |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **Reserved:** Reserved |
| 4 | 0x1<br>RO | **Capabilities List Enable (CAPE):** This bit indicates the presence of an Extended capabilities list items. |
| 3 | 0x0<br>RO/V | **Interrupt Status (INTS):** When set, this bit indicates that an INTx emulation interrupt is pending internally in this function. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 47.2.5 CCR—Offset 9h

The CCR register has an attribute of RW-L PRST to allow configuration software to modify and lock the register from further update. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL').

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] + 9h
**Offset:** [B:0, D:22, F:0] + 9h
**Offset:** [B:0, D:23, F:0] + 9h

**Default:** 00060400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x6<br>RW/L | **Base Class (BC):** The value of 06h indicates that this is a bridge device.<br><br>**Power Well:** PRST |
| 15:8 | 0x4<br>RW/L | **Sub-Class (SC):** This 8-bit value indicates that this device is a PCI-to-PCI Bridge.<br><br>**Power Well:** PRST |
| 7:0 | 0x0<br>RW/L | **Register-Level Programming Interface (RLPI):** This bit indicates that this device is standard (non-subtractive) PCI-to-PCI Bridge.<br><br>**Power Well:** PRST |

## 47.2.6 CLS—Offset Ch

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + Ch
**Offset:** [B:0, D:22, F:0] + Ch
**Offset:** [B:0, D:23, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Cache Line Size (CLS):** These bits specify the system cache-line size in units of DWords. This field is implemented by PCI Express devices but has no effect on device behavior. |

## 47.2.7 IOBASE—Offset 1Ch

The I/O Base and I/O Limit registers define an address range that is used by the PCI-Express port to determine when to forward I/O transactions from one interface to the other using the following formula: IO_BASE (= A[15:12](=IO_LIMIT Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0. The bottom of the defined I/O address range will be aligned to a 4KB boundary while the top of the region specified by IO_LIMIT will be one less than a 4 KB multiple.

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + 1Ch
**Offset:** [B:0, D:22, F:0] + 1Ch
**Offset:** [B:0, D:23, F:0] + 1Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW | **I/O Base Address Bits (IOBA):** These bits define the bottom address of an address range to determine when to forward I/O transactions from one interface to another.<br>These bits correspond to address lines[15:12] for 4 KB alignment. Bits[11:0] are assumed to be 000h. |
| 3:0 | 0x0<br>RO | **I/O Base Addressing Capability (IOBC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only. |

## 47.2.8 IOLIMIT—Offset 1Dh

The I/O Base and I/O Limit registers define an address range that is used by the PCI-Express bridge to determine when to forward I/O transactions from one interface to the other using the following formula: IO_BASE (= A[15:12] (=IO_LIMIT Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] of the IO limit register is treated as FFFh.

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + 1Dh
**Offset:** [B:0, D:22, F:0] + 1Dh
**Offset:** [B:0, D:23, F:0] + 1Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW | **I/O Limit Address Bits (IOLA):** These bits define the top address of an address range to determine when to forward I/O transactions from PCI Express to PCI.<br>These bits correspond to address lines[15:12] for 4 KB aligned window. Bits[11:0] are assumed to be FFFh. |
| 3:0 | 0x0<br>RO | **I/O Limit Addressing Capability (IOLC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only. |

## 47.2.9 MEMBASE—Offset 20h

The Memory Base and Memory Limit registers define a memory mapped I/O non-prefetchable address range (32-bit addresses) and direct the accesses in this range to the PCI-Express port based on the following formula: MEMORY_BASE (= A[31:20] (= MEMORY_LIMIT The upper 12 bits of both the Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, AD[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 20h
**Offset:** [B:0, D:22, F:0] + 20h
**Offset:** [B:0, D:23, F:0] + 20h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Memory Base (MB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range.<br>The incoming address must be greater than or equal to this value. |
| 3:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 47.2.10 MEMLIMIT—Offset 22h

This register controls the processor to PCI-Express non-prefetchable memory access routing based on the following formula as described above: MEMORY_BASE (= A[31:20] The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. The bottom 4 bits of this register are read-only and return zeroes when read. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh. Memory range covered by MEMBASE and MEMLIM registers, are used to map non-prefetchable PCIExpress address ranges (typically where control/ status memory-mapped I/O data structures reside) and PFBASE and PFLIMIT are used to map prefetchable address ranges.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 22h
**Offset:** [B:0, D:22, F:0] + 22h
**Offset:** [B:0, D:23, F:0] + 22h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Memory Limit (ML):** These bits are compared with bits[31:20] of the incoming address to determine the upper limit of the range of memory accesses that will be passed by the port.<br>The incoming address must be less than or equal to this value as described in the formula in 'Memory Base Register' (MEMBASE). |
| 3:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 47.2.11 PFBASE—Offset 24h

The Prefetchable Memory Base and Memory Limit registers define a memory mapped I/O prefetchable address range (32-bit addresses) which is used by the PCI-Express bridge to determine when to forward memory transactions based on the following formula: PREFETCH_MEMORY_BASE (= A[31:20] The upper 12 bits of both the Prefetchable Memory Base and Memory Limit registers are read/ write and corresponds to the upper 12 address bits, A[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, A[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, A[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 24h
**Offset:** [B:0, D:22, F:0] + 24h
**Offset:** [B:0, D:23, F:0] + 24h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Prefetchable Memory Base (PMB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range.<br>The incoming address must be greater than or equal to this value. |
| 3:0 | 0x1<br>RO | **Prefetchable Memory Base Address Capability (PFMCAP):**<br>• 0: 32-bit Prefetchable Memory addressing<br>• 1: 64-bit Prefetchable Memory addressing<br><br>This field indicates that 64-bit addressing is supported for the limit. |

## 47.2.12 PFLIMIT—Offset 26h

This register controls the processor to PCI-Express prefetchable memory access routing based on the following formula as described above: PREFETCH_MEMORY_BASE (= A[31:20] The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 26h
**Offset:** [B:0, D:22, F:0] + 26h
**Offset:** [B:0, D:23, F:0] + 26h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Prefetchable Memory Limit (PML):** These bits are compared with bits[31:20] of the incoming address to determine the upper limit of the prefetchable memory range.<br>The incoming address must be less than or equal to this value as described in the formula in 'Prefetchable Memory Limit Register (PFLIMIT). |
| 3:0 | 0x1<br>RO | **Prefetchable Memory Limit Address Capability (PMLC):** This field indicates that 64-bit addressing is supported for the limit. |

## 47.2.13 PFBASEU—Offset 28h

The Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers are extensions to the Prefetchable Memory Base and Prefetchable Memory Limit registers. If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers specify the upper 32-bits, corresponding to A[63:32], of the 64-bit base and limit addresses which specify the prefetchable memory address range. PREFETCH_MEMORY_BASE_UPPER (= A[63:32] PREFETCH_MEMORY_BASE_LIMIT_UPPER

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] + 28h
**Offset:** [B:0, D:22, F:0] + 28h
**Offset:** [B:0, D:23, F:0] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Prefetchable Memory Base Upper Portion (PMBU):** Corresponds to A[63:32] of the memory address that maps to the upper base of the prefetchable range of memory accesses that will be passed by the PCI-Express bridge. |

## 47.2.14 PFLIMITU—Offset 2Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] + 2Ch
**Offset:** [B:0, D:22, F:0] + 2Ch
**Offset:** [B:0, D:23, F:0] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Prefetchable Memory Limit Upper Portion (PMLU):** Corresponds to A[63:32] of the memory address that maps to the upper limit of the prefetchable range of memory accesses that will be passed by the PCI-Express bridge.<br>The incoming address must be less than or equal to this value as described in the formula in 'Prefetchable Memory Base Upper 32 Bits Register' (PFBASEU). |

## 47.2.15 CAPPTR—Offset 34h

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + 34h
**Offset:** [B:0, D:22, F:0] + 34h
**Offset:** [B:0, D:23, F:0] + 34h

**Default:** 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CPTR):** Contains the offset of the first item in the list of capabilities. (EXPCAPLST) |

## 47.2.16 INTL—Offset 3Ch

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + 3Ch
**Offset:** [B:0, D:22, F:0] + 3Ch
**Offset:** [B:0, D:23, F:0] + 3Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information. |

## 47.2.17 INTP—Offset 3Dh

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] + 3Dh
**Offset:** [B:0, D:22, F:0] + 3Dh
**Offset:** [B:0, D:23, F:0] + 3Dh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RW/L | **Interrupt Pin (INTP):** This register tells which interrupt pin the function uses.<br>• 01h: Generate INTA<br>• 02h: Generate INTB<br>• 03h: Generate INTC<br>• 04h: Generate INTD<br>• Others: Reserved<br><br>**Power Well:** PRST |

## 47.2.18 BCTL—Offset 3Eh

This register provides extensions to the PCI Command Register (PCICMD) that are specific to a bridge. The Bridge Control Register provides many of the same controls for the secondary interface that are provided by the Command Register for the primary interface. Some bits affect operation of both interfaces of the bridge.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] + 3Eh
**Offset:** [B:0, D:22, F:0] + 3Eh
**Offset:** [B:0, D:23, F:0] + 3Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11 | 0x0<br>RO | **Discard Timer SERR# Enable (DTSE):** Not applicable to PCI Express. Hardwired to 0. |
| 10 | 0x0<br>RO | **Discard Timer Status (DTS):** Not applicable to PCI Express. Hardwired to 0. |
| 9 | 0x0<br>RO | **Secondary Discard Timer (SDT):** Not applicable to PCI Express. Hardwired to 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8 | 0x0<br>RO | **Primary Discard Timer (PDT):** Not applicable to PCI Express. Hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not applicable to PCI Express. Hardwired to 0. |
| 6 | 0x0<br>RW | **Secondary Bus Reset (SBR):** Setting this bit triggers a hot reset on the downstream link for the corresponding PCIe Express port and the PCI Express hierarchy domain subordinate to the port. Software must ensure a minimum reset duration of 1us as defined in the PCI Local Bus Specification, Revision 3.0. Hardware will continue to maintain the hot reset state as long as the SBR bit is set. For Root Ports, it is recommended that software assert this field for a minimum of 2 ms to ensure that all downstream links enters hot reset state. A secondary bus reset will not reset any register of a Type 1 configuration space header function. |
| 5 | 0x0<br>RO | **Master Abort Mode (MAM):** Not applicable to PCI Express. Hardwired to 0. |
| 4 | 0x0<br>RW | **VGA 16-bit Decode (VGA16BD):** This bit enables the bridge to provide 16-bit decoding of VGA I/O address precluding the decoding of VGA alias addresses every 1 KB. This bit requires the VGA enable bit (bit 3 of this register) to be set to 1.<br>• 0: execute 10-bit address decode on VGA I/O accesses<br>• 1: execute 16-bit address decode on VGA I/O accesses |
| 3 | 0x0<br>RW | **VGA Enable (VGAE):** This bit modifies the response to VGA-compatible addresses.<br>When set to 1b, the bridge positively decodes and forwards the following transactions from primary side to secondary side regardless of the value of the I/O base and limit registers.<br>The transactions are qualified by the memory enable and I/O enable in the command register.<br>Memory addresses:<br>• 000A 0000h-000B FFFFh<br>I/O addresses:<br>• 3B0h-3BBh and 3C0h-3DFh in first 64 KB of I/O address space (Inclusive of ISA address aliases when IO address bits[15:10] are not decoded) |
| 2 | 0x0<br>RW | **ISA Enable (IE):** This bit modifies the response by the bridge to ISA I/O addresses.<br>This field applies only to I/O addresses that are enabled by the I/O base and I/O limit registers and are in the first 64 KB of PCI I/O space.<br>When this bit is set, the bridge blocks all forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1 KB block (offsets 100h to 3FFh).<br>In the opposite direction (secondary to primary), I/O transactions will be forwarded if they address the last 768B in each 1 KB block.<br>• 1: Forward upstream ISA I/O addresses in the address range defined by the I/O Base and I/O Limit registers that are in the firsts 64KB of PCI I/O address space (Top 768B of each 1K block).<br>• 0: Forward downstream all I/O addresses in the address range defined by the I/O Base and I/O Limit registers. |
| 1 | 0x0<br>RW | **SERR# Enable (SE):** This bit controls the forwarding of PCI Express ERR_COR, ERR_NONFATAL and ERR_FATAL messages to the primary side.<br>• 1: Enables forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL messages.<br>• 0: Disables forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL messages. |
| 0 | 0x0<br>RW | **Parity Error Response Enable (PERE):** This bit controls the response to poisoned TLPs in the PCI Express port.<br>• 1: Enables reporting of poisoned TLP errors.<br>• 0: Disables reporting of poisoned TLP errors. |

## 47.2.19 EXPCAPLST—Offset 40h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +40h
**Offset:** [B:0, D:22, F:0] +40h
**Offset:** [B:0, D:23, F:0] +40h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x80<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list. (PMCAPLST) |
| 7:0 | 0x10<br>RO | **Capability ID (CAPID):** Identifies the function as PCI Express capable. |

## 47.2.20 EXPCAP—Offset 42h

This register stores the version number of the capability item and other base information contained in the capability structure.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +42h
**Offset:** [B:0, D:22, F:0] +42h
**Offset:** [B:0, D:23, F:0] +42h

**Default:** 0042h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13:9 | 0x0<br>RO | **Interrupt Message Number (IMN):** This field indicates the interrupt message number that is generated from the PCI Express port. |
| 8 | 0x0<br>RO | **Slot Implemented (SI):** Hardwired to 0 for non root ports and non DP. |
| 7:4 | 0x4<br>RO | **Device/Port Type (DT):** 4h: Virtual Root Port |
| 3:0 | 0x2<br>RO | **Version Number (VN):** These bits indicate the version number of the PCI Express capability structure. |

## 47.2.21 DEVCAP—Offset 44h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +44h
**Offset:** [B:0, D:22, F:0] +44h
**Offset:** [B:0, D:23, F:0] +44h

**Default:** 00008001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:26 | 0x0<br>RO | **Captured Slot Power Limit Scale (CSPLS):** In combination with the Slot Power Limit value (bits[25:18], this field specifies the upper limit of the power supplied by slot.<br>The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Value field.<br>This value is set by the Set_Slot_Power_Limit message. Does not apply to root ports or integrated devices or end points or downstream ports or Virtual Root Port/switch port |
| 25:18 | 0x0<br>RO | **Captured Slot Power Limit Value (CSPLV):** In combination with the Slot Power Limit Scale value (bits[27:26]), this field specifies the upper limit of the power supplied by slot.<br>The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field.<br>This value is set by the Set_Slot_Power_Limit message. Does not apply to root ports or integrated devices or end points or downstream ports or Virtual Root Port/switch ports |
| 17:16 | 0x0<br>RO | **RSVD1:** Reserved |
| 15 | 0x1<br>RO | **Role-Based Error Reporting (RBER):** The PCIe cluster supports Role-based Error Reporting. |
| 14:12 | 0x0<br>RO | **Undefined:** Reserved. |
| 11:9 | 0x0<br>RO | **Endpoint L1 Acceptable Latency (EPL1AL):** This field only applies to Endpoint and is hardwired to 0 for other types such as root ports, upstream/downstream ports |
| 8:6 | 0x0<br>RO | **Endpoint L0s Acceptable Latency (EPL0AL):** This field only applies to Endpoint and is hardwired to 0 for other types such as root ports, upstream/downstream ports |
| 5 | 0x0<br>RO | **Extended Tag Field Supported (ETFG):**<br>• 0: The PCIe supports 5-bit extended tag<br>• 1: The PCIe supports 8-bit extended tag |
| 4:3 | 0x0<br>RO | **Phantom Functions Supported (PFS):** This field indicates the number of most significant bits of the function number portion of Requester ID in a TLP that are logically combined with the Tag identifier.<br>The PCIe cluster does not support phantom functions |
| 2:0 | 0x1<br>RO/V | **Max Payload size Supported (MPSS):** The PCIe SIP can tolerate upto 512-byte packets as maximum payload for project specific. |

## 47.2.22 DEVCTL—Offset 48h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +48h
**Offset:** [B:0, D:22, F:0] +48h
**Offset:** [B:0, D:23, F:0] +48h

**Default:** 2000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x2<br>RW | **Max_Read_Request_Size (MRRS):** This field sets the maximum Read Requests size of the function as a requester.<br>• 000b: 128 bytes maximum Read Request size<br>• 001b: 256 bytes maximum Read Request size<br>• 010b: 512 bytes maximum Read Request size<br>• 011b: 1024 bytes maximum Read Request size<br>• 100b: 2048 bytes maximum Read Request size<br>• 101b: 4096 bytes maximum Read Request size<br>• Others: Reserved |
| 11 | 0x0<br>RO | **Enable No Snoop (ENOSNP):** If this bit this is set, the function is permitted to set the No Snoop bit in the Requester attributes of transactions it initiates that do not require hardware enforced cache coherency. |
| 10 | 0x0<br>RO | **Auxiliary Power PM Enable (AUXPME):** Not supported |
| 9 | 0x0<br>RO | **Reserved:** Reserved |
| 8 | 0x0<br>RO | **Reserved:** Reserved |
| 7:5 | 0x0<br>RW | **Maximum Payload Size (MPS):** This field sets maximum TLP payload size for the function. As a receiver, the function must handle TLPs as large as the set value. As a Transmitter, the function must not generate TLPs exceeding the set value.<br>• 000b: 128 bytes maximum payload size (default)<br>• 001b: 256 bytes maximum payload size<br>• 010b: 512 bytes maximum payload size<br>• 011b: 1024 bytes maximum payload size<br>• 100b: 2048 bytes maximum payload size<br>• 101b: 4096 bytes maximum payload size<br>• Others: Reserved |
| 4 | 0x0<br>RO | **Reserved:** Reserved |
| 3 | 0x0<br>RW | **Unsupported Request Reporting Enable (URRE):** This bit controls the enabling of ERR_CORR, ERR_NONFATAL or ERR_FATAL messages on PCI Express for reporting 'Unsupported Request' errors. |
| 2 | 0x0<br>RW | **Fatal Error Reporting Enable (FERE):** When this bit is set, generation of the ERR_FATAL message is enabled. |
| 1 | 0x0<br>RW | **NonFatal Error Reporting Enable (NFERE):** When this bit is set, generation of the ERR_NONFATAL message is enabled. |
| 0 | 0x0<br>RW | **Correctable Error Reporting Enable (CERE):** When this bit is set, generation of the ERR_CORR message is enabled. |

## 47.2.23 DEVSTS—Offset 4Ah

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +4Ah
**Offset:** [B:0, D:22, F:0] +4Ah
**Offset:** [B:0, D:23, F:0] +4Ah

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO/V | **Transactions Pending (TP):** When set, this bit indicates that the function has issued Non-Posted Requests that have not been completed. |
| 4 | 0x0<br>RO | **Auxiliary Power Detected (APD):** Auxiliary Power is not supported. |
| 3 | 0x0<br>RW/1C/V | **Unsupported Request Detected (URD):** This bit indicates that this function received an unsupported request from PCI Express link. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 2 | 0x0<br>RW/1C/V | **Fatal Error Detected (FED):** This bit indicates that this function has detected a Fatal error. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 1 | 0x0<br>RW/1C/V | **Non-Fatal Error Detected (NFED):** This bit indicates that this function has detected a Non-Fatal error. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 0 | 0x0<br>RW/1C/V | **Correctable Error Detected (CED):** This bit indicates that this function has detected a Correctable error. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |

## 47.2.24 LINKCAP—Offset 4Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +4Ch
**Offset:** [B:0, D:22, F:0] +4Ch
**Offset:** [B:0, D:23, F:0] +4Ch

**Default:** 06000C11h for D6
**Default:** 16000C11h for D22
**Default:** 17000C11h for D23

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | D6: 0x6<br>D22: 0x16<br>D23: 0x17<br>RW/L | **Port Number (PN):** This field indicates the PCI Express port number assigned to this link for Virtual Root Port/Switch Port.<br><br>**Power Well:** PRST |
| 23:22 | 0x0<br>RO | **RSVD0:** Reserved |
| 21 | 0x0<br>RO | **Link Bandwidth Notification Capability (LBNC):** A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. |
| 20 | 0x0<br>RO | **Data Link Layer Active Error Reporting Capable (DLLERC):** For Upstream Ports and components that do not support this optional capability, this bit must be hardwired to 0b Used device number to distinguish the vRP and vSP for default values |
| 19 | 0x0<br>RO | **Surprise Down Error Reporting Capable (SLDERC):** For Upstream Ports and components that do not support this optional capability, this bit must be hardwired to 0b. |
| 18 | 0x0<br>RO | **Clock Power Management Capable (CPMC):** For Downstream Ports, this bit must be hardwired to 0b. |
| 17:15 | 0x0<br>RO | **L1 Exit Latency (L1EL):** This field indicates the L1 exit latency for the given PCI-Express Link. It indicates the length of time this port requires to complete transition from L1 to L0.<br>• 000: Less than 1us<br>• 001: 1us to less than 2us<br>• 010: 2us to less than 4us<br>• 011: 4us to less than 8us<br>• 100: 8us to less than 16us<br>• 101: 16us to less than 32us<br>• 110: 32us to less than 64us<br>• 111: More than 64us<br><br>If L1 is not supported, the value is undefined |
| 14:12 | 0x0<br>RO | **L0s Exit Latency (L0sEL):** This field indicates the L0s exit latency for the given PCI-Express Link. It indicates the length of time this port requires to complete transition from L0s to L0.<br>• 000b: Less than 64ns<br>• 001b: 64ns to less than 128ns<br>• 010b: 128ns to less than 256ns<br>• 011b: 256ns to less than 512ns<br>• 100b 512 ns to less than 1us<br>• 101b: 1us to less than 2us<br>• 110b: 2us to less than 4us<br>• 111b: More than 4us<br><br>If L0s is not supported, the value is undefined |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11:10 | 0x3<br>RO | **ASPM Support (ASPMSUP):** This field indicates the level of ASPM supported on the given PCI Express Link.<br>• 00b: No ASPM Support<br>• 01b: L0s Supported<br>• 10b: L1 Supported<br>• 11b: L0s and L1 Supported |
| 9:4 | 0x1<br>RO | **Maximum Link Width (MLW):**<br>This field indicates the maximum link width implemented by the given PCI Express Link.<br>• 00h: Reserved<br>• 01h: x1<br>• 02h: x2<br>• 04h: x4<br>• 08h: x8<br>• 10h: x16<br>• 20h: x32 (Unsupported)<br>• Others Reserved |
| 3:0 | 0x1<br>RO | **Maximum Link Speed (MLS):**<br>This field indicates the maximum supported link speed(s) of the associated port.<br>• 0001b: 2.5 Gb/s link speed is supported<br>• 0010b: 5.0 Gb/s and 2.5 Gb/s link speed supported<br>• 0011b: 8.0 Gb/s and 5.0 and 2.5 Gb/s link speed supported<br>• Others: Reserved. |

## 47.2.25 LINKCTL—Offset 50h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +50h
**Offset:** [B:0, D:22, F:0] +50h
**Offset:** [B:0, D:23, F:0] +50h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11 | 0x0<br>RO | **Reserved:** Reserved |
| 10 | 0x0<br>RO | **Link Bandwidth Management Interrupt Enable (LBMIE):** When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. |
| 9 | 0x0<br>RO | **Reserved:** Reserved |
| 8 | 0x0<br>RW/L | **Enable Clock Power Management (ECPM):** Applicable only for Upstream Ports and with form factors that support a 'Clock Request' (CLKREQ#) mechanism, this bit operates as follows:<br>• 0b Clock power management is disabled and device must hold CLKREQ# signal low.<br>• 1b When this bit is Set, the device is permitted to use CLKREQ# signal to power manage Link clock according to protocol defined.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RW | **Extended Synch (ES):** When set, this bit forces extended transmission of 4096 FTS ordered sets in FTS and an extra 1024 TS1 at exit from L1 prior to entering L0. |
| 6 | 0x0<br>RW | **Common Clock Configuration (CCCFG):** When set, this bit indicates that this component and the component at the opposite end of this link are operating with distributed common reference clocks. A value of 0b indicates that this component and the component at the opposite end of this link are operating with asynchronous reference clock. |
| 5 | 0x0<br>RO | **Retrain Link (RL):** When set, this bit initiates link retraining by directing the physical layer LTSSM to recovery state. For the upstream port or virtual switch port, it is Read-only. |
| 4 | 0x0<br>RW | **Link Disable (LD):** This bit is reserved on Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches. |
| 3 | 0x0<br>RO | **Read Completion Boundary (RCB):**<br>This bit indicates the RCB value for Root Port, Endpoints and Bridges.<br>• 0b: 64 byte<br>• 1b: 128 byte |
| 2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1:0 | 0x0<br>RW | **ASPM Control (ASPMCTL):** This field controls the level of ASPM enabled on a given PCI Express Link.<br>• 00b: Disabled<br>• 01b: L0s Entry Enabled<br>• 10b: L1 Entry Enabled<br>• 11b: L0s and L1 Entry Enabled<br>''L0s Entry Enabled' enables the Transmitter to enter L0s. |

## 47.2.26 LINKSTS—Offset 52h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +52h
**Offset:** [B:0, D:22, F:0] +52h
**Offset:** [B:0, D:23, F:0] +52h

**Default:** 1011h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Link Autonomous Bandwidth Status (LABS):** This bit is set by hardware. |
| 14 | 0x0<br>RO | **Link Bandwidth Management Status (LBMS):** This bit is set by hardware. This bit must be set if the Physical Layer reports a non-autonomous speed or width change initiated by the Upstream Port. |
| 13 | 0x0<br>RO | **Data Link Layer Link Active (DLLLA):** This bit indicates the status of the Data Link Control and Management Status Machine. It returns a 1b to indicate the DL_Active state, 0b otherwise. |
| 12 | 0x1<br>RW/L | **Slot Clock Configuration (SCC):** When to 1b, this bit indicates that the component uses the same physical reference clock that he platform provides on the connector.<br>If the device uses an independent clock irrespective of the presence of a reference on the connector, this bit must be clear.<br>• 1: Indicates same reference clock.<br>• 0: Indicates independent reference clock<br><br>**Power Well:** PRST |
| 11 | 0x0<br>RO | **Link Training (LT):** This bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state, or a 1b was written to the Retrain Link bit but the Link training has not yet begun. Hardware clears this bit when the LTSSM exits the Configuration/Recovery state. |
| 10 | 0x0<br>RO | **Undefined:** Not applicable |
| 9:4 | 0x1<br>RO | **Negotiated Link Width (NLW):** This field indicates the negotiated width of the PCI Express link.<br>• 00 0001b: x1<br>• 00 0010b: X2<br>• 00 0100b: x4<br>• 00 1000b: X8<br>• 00 1100b: X12 not supported<br>• 01 0000b: X16<br>• 10 0000b: X32 not supported<br>• All other values are reserved.<br><br>The value in this field is undefined when the link is not up. |
| 3:0 | 0x1<br>RO | **Current Link Speed (CLS):**<br>This field indicates the negotiated link speed of the given PCI Express link.<br>• 0001b: 2.5 Gb/s PCI Express Link<br>• 0010b: 5.0 Gb/s PCI Express Link<br>• 0011b: 8.0 Gb/s PCI Express Link<br>• Others: Reserved |

## 47.2.27 SLOTSTS—Offset 5Ah

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +5Ah
**Offset:** [B:0, D:22, F:0] +5Ah
**Offset:** [B:0, D:23, F:0] +5Ah

**Default:** 0040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x1<br>RO/V | **Presence Detect State (PDS):** This field conveys the Presence Detect status of an adapter in the slot.<br>• 0: Card/Module/Cable slot empty or Cable Slot occupied but not powered<br>• 1: Card/module Present in slot (powered or unpowered) or cable present and powered on the other end. |
| 5:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 47.2.28 ROOTCTL—Offset 5Ch

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +5Ch
**Offset:** [B:0, D:22, F:0] +5Ch
**Offset:** [B:0, D:23, F:0] +5Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:5 | 0x0<br>RO | **RSVD0:** Reserved |
| 4 | 0x0<br>RO | **CRS Software Visibility Enable (CRSSVE):** This bit, when set, enables the Root Port to return Configuration Retry Status (CRS) Completion status to software.<br>• 1: Enable software to received a CRS status. This allows software to make the decision to re-issue the configuration request or move on and re-issue the request at a later time.<br>• 0: Disable software from receiving a CRS response. The PCIe will wait and re-issue the configuration request until it receives a response other than CRS without notifying software. |
| 3 | 0x0<br>RW | **PME Interrupt Enable (PMEIE):** This field controls the generation of interrupts for PME messages.<br>• 1: Enables interrupt generation upon receipt of a PME message as reflected in the PME Status bit defined in the ROOTSTS register. A PME interrupt is generated if the PMESTATUS register bit defined in Root Status Register (ROOTSTS), is set when this bit is set from a cleared state.<br>0: Disables interrupt generation for PME messages. |
| 2 | 0x0<br>RW | **System Error on Fatal Error Enable (SEFEE):** This field controls generation of system errors in the PCI-Express port hierarchy for fatal errors.<br>• 1: indicates that a System Error should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port.<br>• 0: No System Error should be generated on a fatal error (ERR_FATAL) reported by any of the devices in the hierarchy. |
| 1 | 0x0<br>RW | **System Error on Non-Fatal Error Enable (SENFEE):** This field controls generation of system errors in the PCI-Express port hierarchy for non-fatal errors.<br>• 1: indicates that a System Error should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port.<br>• 0: No System Error should be generated on a non-fatal error (ERR_NONFATAL) reported by any of the devices in the hierarchy. |
| 0 | 0x0<br>RW | **System Error on Correctable Error Enable (SECEE):** This field controls generation of system errors in the PCI-Express port hierarchy for correctable errors.<br>• 1: indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port<br>• 0: No System Error should be generated on a correctable error (ERR_COR) reported by any of the devices in the hierarchy associated with and including this PCI-Express port. |

## 47.2.29 ROOTCAP—Offset 5Eh

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +5Eh
**Offset:** [B:0, D:22, F:0] +5Eh
**Offset:** [B:0, D:23, F:0] +5Eh

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/L | **CRS Software Visibility (CRSSV):** This bit, when set, indicates that the Root Port is capable of returning Configuration Retry Status (CRS) on completions to software. Based on Design input, the CRS notification will be implemented in the EIP PCIe macro and hence the default is set to 1. The field attribute is made as RW-O for any specific changes to be installed by BIOS<br><br>**Power Well:** PRST |

## 47.2.30 ROOTSTS—Offset 60h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +60h
**Offset:** [B:0, D:22, F:0] +60h
**Offset:** [B:0, D:23, F:0] +60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RO/V | **PME Pending (PMEPEND):** This field indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software, the pending PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending.<br><br>***Note:*** The root port can handle two outstanding PM_PME messages in its internal queues of the Power Management controller per port. If the downstream device issues more than 2 PM_PME messages successively, it will be dropped. |
| 16 | 0x0<br>RW/1C/V | **PME Status (PMESTS):** This field indicates status of a PME that is underway in the PCI-Express port. 1: PME was asserted by a requester as indicated by the PMEREQID field This bit is cleared by software by writing a '1'. Subsequent PMEs are kept pending until the PME Status is cleared. |
| 15:0 | 0x0<br>RO/V | **PME Requester ID (PMERID):** This field indicates the PCI requester ID of the last PME requestor. |

## 47.2.31 DEVCAP2—Offset 64h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +64h
**Offset:** [B:0, D:22, F:0] +64h
**Offset:** [B:0, D:23, F:0] +64h

**Default:** 00000020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:18 | 0x0<br>RO | **OBFF Supported (OBFFS):**<br>• 00b: OBFF Not Supported<br>• 01b: OBFF supported using Message signaling only<br>• 10b: OBFF supported using WAKE# signaling only<br>• 11b: OBFF supported using WAKE# and Message signaling Applicable only to Root Ports, Switch Ports, and Endpoints that support this capability. Must be 00b for other function types. |
| 17:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13:12 | 0x0<br>RO | **TPH Completer Supported (TPHCS):** Value indicates Completer support for TPH or Extended TPH.<br>• 00b:TPH and Extended TPH Completer not supported<br>• 01b: TPH Completer supported, Extended TPH completer not supported<br>• 10b: Reserved<br>• 11b: Both TPH and Extended TPH Completer supported Applicable only to Root Ports and Endpoints. Must be 00b for other function types. |
| 11 | 0x0<br>RW/L | **LTR Mechanism Supported (LTRMS):** This bit when set indicates support for the Latency Tolerance Reporting (LTR) mechanism.<br><br>**Power Well:** PRST |
| 10 | 0x0<br>RO | **No RO-enabled PR-PR Passing (NROEPRPASS):** This bit if set to 1b disables passing permitted by row entry A2b in PCIe ordering table that is associated with the Relaxed Ordering Attribute field being set. This bit applies only for Switches and RCs that support peer-to-peer traffic between ports. The bit only applies to posted requests being forwarded through the switch or RC. It does not apply to traffic originating or terminating with in the switch or RC itself. |
| 9 | 0x0<br>RO | **AD128 CAS Completer Supported (AD128ACS):** This bit must be set to 1b if the function supports this optional capability. Applicable to functions with memory space BARs as well as Root Ports. This is not implemented in Virtual Root port |
| 8 | 0x0<br>RO | **AD64-bit AtomicOp Completer Supported (AD64ACS):** This bit must be set to 1b if the function supports this optional capability. Applicable to Functions with Memory Space BARs as well as all Root Ports, must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This is not implemented in Virtual Root port |
| 7 | 0x0<br>RO | **AD32 bit AtomicOp Completer Supported (AD32ACS):** Applicable to Functions with Memory Space BARs as well as all Root Ports, must be 0b otherwise. Includes FetchAdd, Swap, and CAS AtomicOps. This bit must be set to 1b if the Function supports this optional capability. This is not implemented in Virtual Root port |
| 6 | 0x0<br>RW/L | **AtomicOp Routing Supported (ARS):** This bit must be set to 1b if the port supports this optional capability. Applicable only to switch upstream ports, Switch downstream ports, and root ports,. This is not implemented in Virtual Root port.<br><br>**Power Well:** PRST |
| 5 | 0x1<br>RW/L | **ARI Forwarding Supported (ARI):** This bit is set to 1b indicating that the downstream port supports this capability. Applicable only to Switch Downstream Ports and Root Ports, must be 0b for other Function types. This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RO | **Completion Timeout Disable Support (CTDS):** A value of 1b indicates support for the completion Timeout Disable Mechanism. Support of completion timeout disable is optional for Root Ports. The Completion timeout is not supported for vRP/vSP |
| 3:0 | 0x0<br>RO | **Completion Timeout Range Supported (CTRS):** This field indicates device support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of request issues on PCI Express.<br>Four time values ranges are defined:<br>• Range A: 50us to 10ms<br>• Range B: 10ms to 250ms<br>• Range C: 250ms to 4s<br>• Range D: 4s to 64s<br><br>Bits ares set according to table below to show timeout value ranges supported.<br>• 0000b: Completions Timeout programming not supported -- values is fixed by implementation in the range 50us to 50ms.<br>• 0001b: Range A<br>• 0010b: Range B<br>• 0011b: Range A and B<br>• 0110b: Range B and C<br>• 0111b: Range A, B, and C<br>• 1110b: Range B, C, and D<br>• 1111b: Range A, B, C, and D<br>• All other values are reserved.<br><br>For all other devices this field is reserved and maybe be hardwired to 0000b. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL'). CT Ranges are defined for Virtual Root Port / Virtual Switch Port is 0 (unsupported) |

## 47.2.32 DEVCTL2—Offset 68h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +68h
**Offset:** [B:0, D:22, F:0] +68h
**Offset:** [B:0, D:23, F:0] +68h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:13 | 0x0<br>RO | **OBFF Enable (OBFFE):** Reserved. |
| 12:11 | 0x0<br>RO | **RSVD1:** Reserved |
| 10 | 0x0<br>RW/V | **LTR Mechanism Enable (LTRME):** When set to 1b, this bit enables Upstream Ports to send LTR messages and Downstream Ports to process LTR Messages. Applicable to Root Ports, Switches, and Endpoints that implement the LTR capability must implement this bit. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status. |
| 9 | 0x0<br>RO | **IDO Completion Enable (IDOCE):** This bit when set to 1b allows the function to set the ID-Based Ordering (IDO) bit of completions it returns. Applicable only to Endpoints including RC integrated Endpoints and Root Ports. |
| 8 | 0x0<br>RO | **IDO Request Enable (IDORE):** This bit when set to 1b allows the function to set the ID-Based Ordering (IDO) bit on requests it initiates. Applicable only to Endpoints including RC integrated Endpoints and Root Ports. |
| 7 | 0x0<br>RO | **AtomicOp Egress Blocking (AEB):** When this bit is set, AtomicOp requests that target this out going Egress port must be blocked. Applicable and mandatory for switch Upstream Ports, Switch Downstream Ports, and Root Ports that implement AtomicOp routing capability. Otherwise, must be hardwired to 0b. AEB is not supported for Virtual Root port but applicable for Virtual Switch Port if supported. |
| 6 | 0x0<br>RO | **AtomicOp Requester Enable (ARE):** Applicable only to Endpoints and Root Ports, must be hardwired to 0b for other Function types. |
| 5 | 0x0<br>RW | **ARI Forwarding Enable (ARIE):** When set to 1b, ARI is enabled for the downstream port or root ports. The Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port. Default value of this bit is 0b. |
| 4 | 0x0<br>RO | **Completion Timeout Disable (CTD):**<br>• 1: Disable the completions timeout mechanism for all NP transactions.<br>• 0: Completion timeout is enabled for all NP transactions. |
| 3:0 | 0x0<br>RO | **Completion Timeout Value (CTV):** In devices that support completion timeout programmability, this field allows system software to modify the completion timeout range. The following encodings and corresponding timeout ranges are defined based on implementation choice. This field provides finer control based over the ranges chosen by DEVAP2.CTRS.<br>The implementation specific range is shown in brackets<br>• 0000b: 50us to 50ms (16.8ms - 25.2ms based on core clk period)<br>• 0001b: 50us to 100us (65.5us - 99.3us based on core clk period)<br>• 0010b: 1ms to 10ms (4.2ms - 6.3ms based on core clk period)<br>• 0101b: 16ms to 55ms (33.6ms - 50.3ms based on core clk period)<br>• 0110b: 65ms to 210ms (134.2ms - 201.3ms based on core clk period)<br>• 1001b: 260ms to 900ms (536.9ms - 805.3ms based on core clk period)<br>• 1010b: 1s to 3.5s (2.1s - 3.2s based on core clk period)<br>• 1101b: 4s to 13s (8.6s - 12.9s based on core clk period)<br>• 1110b: 17s to 64s (34.4s - 51.5s based on core clk period)<br>• All others are reserved.<br><br>CTV is not supported in UP, DP and vRP/vSP It is highly recommended that the completion timeout value not be less then 10ms. A small completion timeout value may result in premature completion timeout for slower responding devices. |

## 47.2.33 LINKCAP2—Offset 6Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +6Ch
**Offset:** [B:0, D:22, F:0] +6Ch
**Offset:** [B:0, D:23, F:0] +6Ch

**Default:** 00000002h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22:16 | 0x0<br>RW/L | **Lower SKP OS Reception Supported Speeds Vector (LSKPOSRSSV):** If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports receiving SKP OS at the rate defined for SRNS while running in SRIS.<br>Bit definitions within this field are:<br>• Bit 0 2.5 GT/s<br>• Bit 1 5.0 GT/s<br>• Bit 2 8.0 GT/s<br>• Bits 6:3 RsvdP<br><br>Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions. Behavior is undefined if a bit is Set in this field and the corresponding bit is not Set in the Supported Link Speeds Vector. Currently, design does not support reception of low frequency SOS in L0.<br><br>**Power Well:** PRST |
| 15:9 | 0x0<br>RW/L | **Lower SKP OS Generation Supported Speeds Vector (LSKPOSGSSV):** If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports software control of the SKP Ordered Set transmission scheduling rate.<br>Bit definitions within this field are:<br>• Bit 0 2.5 GT/s<br>• Bit 1 5.0 GT/s<br>• Bit 2 8.0 GT/s<br>• Bits 6:3 RsvdP<br><br>Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions. Behavior is undefined if a bit is Set in this field and the corresponding bit is not Set in the Supported Link Speeds Vector.<br><br>**Power Well:** PRST |
| 8 | 0x0<br>RW/L | **Crosslink Supported (CLS):** When set to 1b, this bit indicates that the associated Port supports crosslinks When set to 0b on a Port that supports Link speeds of 8.0 GT/s or higher, this bit indicates that the associated Port does not support crosslinks. When set to 0b on a Port that only supports Link speeds of 2.5 GT/s or 5.0 GT/s, this bit provides no information regarding the Port's level of crosslink support. It is recommended that this bit be Set in any Port that supports crosslinks even though doing so is only required for Ports that also support operating at 8.0 GT/s or higher Link speeds.<br>***Note:*** Software should use this bit when referencing fields whose definition depends on whether or not the Port supports crosslinks.<br><br>**Power Well:** PRST |
| 7:1 | 0x1<br>RO | **Supported Link Speeds Vector (SLNKSPV):** This field indicates the supported Link speed(s) of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported, otherwise, the Link speed is not supported.<br>Bit definitions are:<br>• bit 1 2.5 GT/s<br>• bit 2 5.0 GT/s<br>• bit 3 8.0 GT/s<br>• Others: Reserved |
| 0 | 0x0<br>RO | **RSVD1:** Reserved |

## 47.2.34 LINKCTL2—Offset 70h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +70h
**Offset:** [B:0, D:22, F:0] +70h
**Offset:** [B:0, D:23, F:0] +70h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **Compliance Preset/De-emphasis (CD):** For 8 GT/s and 5 GT/s: This bit sets the transmitter preset level forPolling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b.<br>The Encoding are defined as follows:<br><br>8 GT/s Rate:<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br><br>5 GT/s Rate:<br>• 001b: -3.5dB<br>• 000b: -6 dB<br>• All other are reserved.<br><br>When the link is operating at 2.5 Gb/s, the setting of this bit has no effect. Components that support only 2.5 GT/s speed are permitted to hardwire this bit to 0b. |
| 11 | 0x0<br>RO | **Compliance SOS (CSOS):** When set to 1b, the LTSSM is required to send Skip Ordered Sets periodically in between the (modified) compliance patterns. |
| 10 | 0x0<br>RO | **Enter Modified Compliance (EMC):** When set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value. |
| 9:7 | 0x0<br>RO | **Transmit Margin (TM):** This field controls the value of the non-de-emphasized voltage level at the transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate.<br>• 000b: Normal operating range<br>• 001b: 800-1200mV for full swing and 400-700mV for half-swing<br>• 010b-110b: TBD<br>• Others: Reserved<br><br>This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value. |
| 6 | 0x0<br>RO | **Selectable De-emphasis (SD):** This bit is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and upstream ports of switches. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RO | **Hardware Autonomous Speed Disable (HASD):** When set, this bit disables hardware from changing the link speed for device specific reasons other than attempting to correct unreliable link operation by reducing link speed for device-specific reasons other than attempting to correct unreliable link operations by reducing link speed. Initial transition to the highest supported common link speed is not blocked by this bit. |
| 4 | 0x0<br>RO | **Enter Compliance (EC):** Software is permitted to force a link to enter compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link. |
| 3:0 | 0x1<br>RO | **Target Link Speed (TLS):** For downstream ports and root ports, this field sets an upper limit on link operational speed by restricting the values advertised by the Downstream Port in its training sequences.<br>• 0001b: 2.5 Gb/s Target Link Speed<br>• 0010b: 5.0 Gb/s Target Link Speed<br>• 0011b: 8.0 Gb/s Target Link Speed<br>• Others: Reserved<br><br>The encoding is the binary value of the bit in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the desired target Link speed. All other encodings are reserved If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined. The default value of this field is the highest Link speed supported by the component (as reported in the Max Link Speed field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value. For both Upstream and Downstream Ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode. |

## 47.2.35 LINKSTS2—Offset 72h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +72h
**Offset:** [B:0, D:22, F:0] +72h
**Offset:** [B:0, D:23, F:0] +72h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **Link Equalization Request (LNKEQREQ):** This bit is set by hardware to request the Link equalization process to be performed on the Link. |
| 4 | 0x0<br>RO | **Equalization Phase 3 Successful (EQPH3SUCC):** When set to 1b, this bit indicates that Phase 3 of the Transmitter Equalization procedure has successfully completed. |
| 3 | 0x0<br>RO | **Equalization Phase 2 Successful (EQPH2SUCC):** When set to 1b, this bit indicates that Phase 2 of the Transmitter Equalization procedure has successfully completed. |
| 2 | 0x0<br>RO | **Equalization Phase 1 Successful (EQPH1SUCC):** When set to 1b, this bit indicates that Phase 1 of the Transmitter Equalization procedure has successfully completed. |
| 1 | 0x0<br>RO | **Equalization Complete (EQCMPLT):** When set to 1b, this bit indicates that the Transmitter Equalization procedure has successfully completed. |
| 0 | 0x0<br>RO | **Current De-emphasis Level (CDL):** When the link is operating at 5 Gb/s speed, this bit reflects the level of de-emphasis.<br>• 1b: -3.5 dB<br>• 0b: -6dB<br>The value in this bit is undefined when the Link is not operating at 25.0 GT/s speed. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. |

## 47.2.36 PMCAPLST—Offset 80h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +80h
**Offset:** [B:0, D:22, F:0] +80h
**Offset:** [B:0, D:23, F:0] +80h

**Default:** 8801h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x88<br>RO | **Next Pointer (NextPointer):** Contains the offset of the next item in the capabilities list. (SSCAPLST) |
| 7:0 | 0x1<br>RO | **Capability ID (CapabilityID):** Identifies the function as PCI Power Management capable. |

## 47.2.37 PMCAP—Offset 82h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +82h
**Offset:** [B:0, D:22, F:0] +82h
**Offset:** [B:0, D:23, F:0] +82h

**Default:** C803h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x19<br>RO | **PME_Support (PMES):** PME assertion is supported when in D3hot/D3cold. Identifies power states in which the PCIe can send an 'Assert_PMEGPE/ Deassert PMEGPE' message. Bits 15, 14 and 11 must be set to '1' for PCIPCI bridge structures representing ports on root complexes. The definition of these bits is taken from the PCI Bus Power Management Interface Specification Revision 1.1.<br>• XXXX1b - Assert_PMEGPE/Deassert PMEGPE can be sent from D0<br>• XXX1Xb - Assert_PMEGPE/Deassert PMEGPE can be sent from D1 (Not supported by PCI-E)<br>• XX1XXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D2 (Not supported by PCI-E)<br>• X1XXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 hot (Supported by PCI-E)<br>• 1XXXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 cold (Supported by PCI-E)<br>In the EIP PCIe implementation, there is no PME support for D3 cold as the part does not have the capability. For PCIe compliance purposes, bit 15 is set to 1. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** Not supported |
| 9 | 0x0<br>RO | **D1 Support (D1S):** Not supported |
| 8:6 | 0x0<br>RO | **Auxiliary Current (AC):** Auxiliary power is not supported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Device-specific initialization is not required when transitioning to D0 from D3hot state. This bit is zero. |
| 4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x0<br>RO | **PME Clock (PMECLK):** Does not apply to PCI Express. Hard-wired to 0. |
| 2:0 | 0x3<br>RO | **Version (VER):** The PM implementation in the PCIe cluster is compliant with PCI Bus Power Management Interface Specification, Revision 1.2. |

## 47.2.38 PMCSR—Offset 84h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +84h
**Offset:** [B:0, D:22, F:0] +84h
**Offset:** [B:0, D:23, F:0] +84h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V/P | **PME Status (PMESTS):** Reserved. |
| 14:13 | 0x0<br>RO | **Data Scale (DC):** Not supported |
| 12:9 | 0x0<br>RO | **Data Select (DS):** Not supported |
| 8 | 0x0<br>RW/P | **PME Enable (PMEEN):** Gates assertion of the PME message. |
| 7:4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x1<br>RW/L | **No Soft Reset (NSR):** This bit when 1b indicates that a device transitioning from D3hot to D0 does not perform an internal reset. The configuration context is preserved.<br><br>**Power Well:** PRST |
| 2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1:0 | 0x0<br>RW/V | **Power State (PS):** This field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the supported values is given below:<br>• 0h - D0<br>• 3h - D3hot<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. |

## 47.2.39 PMBSE—Offset 86h

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] +86h
**Offset:** [B:0, D:22, F:0] +86h
**Offset:** [B:0, D:23, F:0] +86h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Bus Power/Clock Control Enable (BPCC_EN):** Neither bus or clock control of PCI is supported when in D3hot state. This bit is hard-wired to 0. |
| 6 | 0x0<br>RO | **B2/B3# (B23EN):** Not supported. This bit has no meaning since the BPCC_En bit is hard-wired to 0. |
| 5:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 47.2.40 SSCAPLST—Offset 88h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +88h
**Offset:** [B:0, D:22, F:0] +88h
**Offset:** [B:0, D:23, F:0] +88h

**Default:** 900Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x90<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list. (MSICAPLST) |
| 7:0 | 0xd<br>RO | **Capability ID (CAPID):** Identifies the function as Subsystem Identification capable. |

## 47.2.41 SSVID—Offset 8Ch

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +8Ch
**Offset:** [B:0, D:22, F:0] +8Ch
**Offset:** [B:0, D:23, F:0] +8Ch

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RW/L | **Subsystem Vendor Identifier (SSVID):** Assigned by PCI-SIG for vendor ID.<br><br>**Power Well:** PRST |

## 47.2.42 SSID—Offset 8Eh

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +8Eh
**Offset:** [B:0, D:22, F:0] +8Eh
**Offset:** [B:0, D:23, F:0] +8Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **Subsystem Identifier (SSID):** Assigned to uniquely identify the subsystem vendor.<br><br>**Power Well:** PRST |

## 47.2.43 MSICAPLST—Offset 90h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +90h
**Offset:** [B:0, D:22, F:0] +90h
**Offset:** [B:0, D:23, F:0] +90h

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list (MSIXCAPLST). A null value is used to indicate that this is the last capability. The Next Pointer is terminated for DP, vRP/vSP, RP and DMI EP in RC. |
| 7:0 | 0x5<br>RO | **Capability ID (CAPID):** Identifies the function as MSI capable. |

## 47.2.44 MSICTL—Offset 92h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +92h
**Offset:** [B:0, D:22, F:0] +92h
**Offset:** [B:0, D:23, F:0] +92h

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x1<br>RW/L | **Per-vector masking capable (PVM):** This bit indicates that PCI Express ports support MSI per-vector masking.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **Address 64-Bit Capable (AD64C):** When set, this bit indicates that the function is capable of generating a 64-bit message address. |
| 6:4 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Only one message is supported. These bits are R/W for software compatibility. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Only one message is supported. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. |

## 47.2.45 MSIADDR—Offset 94h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +94h
**Offset:** [B:0, D:22, F:0] +94h
**Offset:** [B:0, D:23, F:0] +94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address:** Message address specified by the system, always DWORD aligned |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 47.2.46 MSIDATA—Offset 98h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +98h
**Offset:** [B:0, D:22, F:0] +98h
**Offset:** [B:0, D:23, F:0] +98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>• 0: Edge Triggered<br>• 1: Level Triggered |
| 14 | 0x0<br>RW | **Level (LVL):** if TM is 0h, then this field is a don't care. Edge triggered messages are always treated as assert messages.<br>For level triggered interrupts, this bit reflects the state of the interrupt input if TM is 1h, then<br>• 0: Deassert Messages<br>• 1: Assert Messages |
| 13:12 | 0x0<br>RW | **Future:** These bits are don't care for an IOxAPIC interrupt message data field specification. |
| 11:8 | 0x0<br>RW | **Delivery Mode (DM):**<br>• 0000: Fixed: Trigger Mode can be edge or level.<br>• 0001: Lowest Priority: Trigger Mode can be edge or level.<br>• 0010: SMI/PMI/MCA - Not supported via MSI of root port<br>• 0011: Reserved - Not supported via MSI of root port<br>• 0100: NMI - Not supported via MSI of root port<br>• 0101: INIT - Not supported via MSI of root port<br>• 0110: Reserved<br>• 0111: ExtINT - Not supported via MSI of primary port<br>• 1000 -1111 - Reserved |
| 7:0 | 0x0<br>RW | **Interrupt Vector (IV):** The interrupt vector will be modified by the EIP PCIe Gen3 to provide context sensitive interrupt information for different events that require attention from the processor. |

## 47.2.47 MSIMSK—Offset 9Ch

The MSI Mask Bit register enables software to disable message sending on a per-vector basis.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +9Ch
**Offset:** [B:0, D:22, F:0] +9Ch
**Offset:** [B:0, D:23, F:0] +9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RW | **Mask Bits (MSKB):** For each Mask bit that is set, the PCI Express port is prohibited from sending the associated message. Corresponding bits are masked if set to '1' |

## 47.2.48 MSIPENDING—Offset A0h

The Mask Pending register enables software to defer message sending on a per-vector basis.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +A0h
**Offset:** [B:0, D:22, F:0] +A0h
**Offset:** [B:0, D:23, F:0] +A0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RO/V | **Pending Bits (PB):** For each Pending bit that is set, the PCI Express port has a pending associated message. Corresponding bits are pending if set to '1' |

## 47.2.49 RPPMCSR—Offset A8h

bek_eip_pcie3_regs_usx16.fm bek_eip_pcie3_regs_usx8.fm bek_eip_pcie3_regs_vsp.fm cnx_eip_pcie3_regs_dmix4.fm cnx_eip_pcie3_regs_rp4px4.fm cnx_eip_pcie3_regs_rp4px16.fm cnx_eip_pcie3_regs_ep.fm cnx_eip_pcie3_regs_rldsx16.fm cdf_eip_pcie3_regs_rlusx16.fm dnv_eip_pcie3_regs_rp4px8.fm dnv_eip_pcie3_regs_rp4px16.fm dnv_eip_pcie3_regs_rp4px4.fm dnv_eip_pcie3_regs_vrp.fm

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +A8h
**Offset:** [B:0, D:22, F:0] +A8h
**Offset:** [B:0, D:23, F:0] +A8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3:2 | 0x0<br>RW | **PME_TO_ACK Timeout (PME_TO_ACK_TO):** This timer is activated from the time the PME_Turn_Off message is sent to the link. It will be reset once the timer expires or when PME_TO_ACK has been received from the downstream device and link has been put into L2/L3 Ready state. Once the timer expires, the Sx sequence will continue to take the link down.<br>• 00: 1 ms (default)<br>• 01: 10 ms<br>• 10: 1 us (Pre-Si)<br>• 11: Disable |
| 1:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 47.2.50 PMCSRS—Offset B0h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +B0h
**Offset:** [B:0, D:22, F:0] +B0h
**Offset:** [B:0, D:23, F:0] +B0h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/P | **PME Status (PMESTS):** This field is utilized for Software to restore the PME Status once exit Sx State. Hardware will reflect the value to the associated PMCSR.PMESTS if this bit was written by Software. |
| 14:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 47.2.51 SMICSR—Offset C8h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +C8h
**Offset:** [B:0, D:22, F:0] +C8h
**Offset:** [B:0, D:23, F:0] +C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18 | 0x0<br>RW/1C/V | **Hot Plug DLLSC SMI Status (HPLSCSMISTS):** This bit is set when rising edge detected as result of both SLOTSTS.DLLSCS and SMICSR.HPSMIE being set. This field will be treated as reserved for vRP. |
| 17 | 0x0<br>RW/1C/V | **Hot Plug SMI Status (HPSMISTS):** This bit is set when rising edge detected as result of both SLOTSTS.PDCS and SMICSR.HPSMIE being set. This field will be treated as reserved for vRP. |
| 16 | 0x0<br>RW/1C/V | **Power Management SMI Status (PMSMISTS):** This bit is set when rising edge detected as result of both ROOTSTS.PMESTS and SMICSR.PMSMIE being set. |
| 15:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Hot Plug SMI Enable (HPSMIE):** Enables the Root Port/DSP to generate SMI whenever a Hot Plug event is detected. This field will be treated as reserved for vRP. |
| 0 | 0x0<br>RW | **Power Management SMI Enable (PMSMIE):** Enables the Root Port to generate SMI whenever a power management event is detected. |

## 47.2.52 SCICSR—Offset CCh

The Tag field is being used to differentiate between PMSCI (Tag = '001') and HPSCI (Tag = '000') for PMC whenever a single set of Assert/Deassert SCI is sent.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +CCh
**Offset:** [B:0, D:22, F:0] +CCh
**Offset:** [B:0, D:23, F:0] +CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18 | 0x0<br>RW/1C/V | **Hot Plug DLLSC SCI Status (HPLSCSCISTS):** This bit is set when rising edge detected as result of both SLOTSTS.DLLSCS and SCICSR.HPSCIE being set. This field will be treated as reserved for vRP. |
| 17 | 0x0<br>RW/1C/V | **Hot Plug SCI Status (HPSCISTS):** This bit is set when rising edge detected as result of both SLOTSTS.PDCS and SCICSR.HPSCIE being set. This field will be treated as reserved for vRP. |
| 16 | 0x0<br>RW/1C/V | **Power Management SCI Status (PMSCISTS):** This bit is set when rising edge detected as result of both ROOTSTS.PMESTS and SCICSR.PMSCIE being set. |
| 15:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Hot Plug SCI Enable (HPSCIE):** Enables the Root Port/DSP to generate SCI whenever a Hot Plug event is detected. This field will be treated as reserved for vRP. |
| 0 | 0x0<br>RW | **Power Management SCI Enable (PMSCIE):** Enables the Root Port to generate SCI whenever a power management event is detected. |

## 47.2.53 PBTXNCTL—Offset D8h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +D8h
**Offset:** [B:0, D:22, F:0] +D8h
**Offset:** [B:0, D:23, F:0] +D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RW/P | **ECO31:** These register fields are of type 'RWS' and have no associated functionality currently. They are allocated for future use to add defeature capability as the need arise during tapeout/ECO netlist fixes. |
| 17:0 | 0x0<br>RW/P | **ECO17:** These register fields are of type 'RWS' and have no associated functionality currently. They are allocated for future use to add defeature capability as the need arise during tapeout/ECO netlist fixes. |

## 47.2.54 PLKCTL—Offset EAh

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +EAh
**Offset:** [B:0, D:22, F:0] +EAh
**Offset:** [B:0, D:23, F:0] +EAh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW/L | **Lnk Layer Capability Lock (LLCL):** Lock key bit for all RWS-L bits located in the following CSRs. VCxRCTL XPLDEF XPLDEF2 LLRBERRINJCTL RXPERRINJCTL CRCERRINJCTL CRCERRINJMSK RTRYCTL PACKTHRESH<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **Txn Layer Capability Lock (TLCL):** Lock key bit for all RWS-L bits located in XPEINJ[1:0], XPCDTHROTTLEP,N,C, XPTDEF*, MBAR01CTL EXPPTMBARCTL.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RWS-L bits (capabilities, next capability pointer, SSID/SVID, slot register, etc) bits for the function.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |

## 47.2.55 LTRCSR—Offset ECh

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +ECh
**Offset:** [B:0, D:22, F:0] +ECh
**Offset:** [B:0, D:23, F:0] +ECh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits of LTRCSR and LTROVR register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 30:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RW/1C/V | **Invalid LTR Latency Scale Value Received (IVLSVR):** Set to '1' by hardware when an LTR message is received with Not Permitted Latency Scale Value and both the LTR Mechanism Supported bit and the LTR Mechanism Enable bit are set. Cleared to '0' by software writing 1 to this register. |
| 16 | 0x0<br>RW/1C/V | **LTR Received (LTRR):** Set to '1' by hardware when an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable bit are set. It can be set/cleared by hardware internally, or software by writing 1 to clear it. |
| 15:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3 | 0x0<br>RW/P | **LTR Aggregation Mode Disable (LTRAMD):**<br>• Default is 0 to enable LTR aggregation mode.<br>• Set 1 to disable LTR Aggregation Mode. |
| 2 | 0x0<br>RW/L | **LTR Override Policy (LTROVRPLCY):** This register bit defines the LTR override behavior when the respective LTR Override Enable bits are set. 0: The LTR Override values will be in effect when the link is up and in D0, until a valid new LTR message is received from the device. 1: The LTR Override values will be in effect when the link is up and in D0. LTR values from the valid new LTR message received from device will be ignored. Note: This register should be programmed prior to enabling LTR override and should not be modified after LTR override has been enabled.<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **LTR No-Snoop Override Enable (LTRNSOVREN):** When this bit is set, the latency tolerance values for this port will be overridden to the values programmed in the LTRNSLOVRV, LTRNSLSOVRV and LTRNSROVR fields. The latency values from the LTR messages received from the devices will be ignored. Hardware should treat this bit as dynamic and the transition of this bit from '0' to '1' will cause a new IOSF sideband LTR message to be sent to update PMC if the new aggregated latency values change as a result of this.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **LTR Snoop Override Enable (LTRSOVREN):** When this bit is set, the latency tolerance values for this port will be overridden to the values programmed in the LTRSLOVRV, LTRSLSOVRV and LTRSROVR fields. The latency values from the LTR messages received from the devices will be ignored. Hardware should treat this bit as dynamic and the transition of this bit from '0' to '1' will cause a new IOSF sideband LTR message to be sent to update PMC if the new aggregated latency values change as a result of this.<br><br>**Power Well:** PRST |

## 47.2.56 LTRL—Offset F0h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +F0h
**Offset:** [B:0, D:22, F:0] +F0h
**Offset:** [B:0, D:23, F:0] +F0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO/V | **LTR No-Snoop Latency Field (LTRNSLF):** This field contains the No-Snoop Latency field of the latest received LTR Message. When an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable field are set, bytes 12:13 of the PCIe header are stored in this register. This includes the case where the Latency Scale Value contains 'Not Permitted' encoding. |
| 15:0 | 0x0<br>RO/V | **LTR Snoop Latency Field (LTRSLF):** This field contains the Snoop Latency field of the latest received LTR Message. When an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable field are set, bytes 14:15 of the PCIe header are stored in this register. This includes the case where the Latency Scale Value contains 'Not Permitted' encoding. |

## 47.2.57 LTROVR—Offset F4h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +F4h
**Offset:** [B:0, D:22, F:0] +F4h
**Offset:** [B:0, D:23, F:0] +F4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **LTR No-Snoop Requirement Bit Override (LTRNSROVR):** This field contains the No-Snoop Requirement bit override value for this particular PCIe root port. This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 30:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:26 | 0x0<br>RW/L | **LTR No-Snoop Latency Scale Override Value (LTRNSLSOVRV):** This field contains the Snoop latency scale override value for this particular PCIe root port.<br>000: LTRNSLOVRV times 1 ns<br>001: LTRNSLOVRV times 32 ns<br>010: LTRNSLOVRV times 1,024 ns<br>011: LTRNSLOVRV times 32,768 ns<br>100: LTRNSLOVRV times 1,048,576 ns<br>101: LTRNSLOVRV times 33,554,432 ns<br>Others: Not Permitted.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 25:16 | 0x0<br>RW/L | **LTR No-Snoop Latency Override Value (LTRNSLOVRV):** This field contains the No-Snoop latency override value for this particular PCIe root port. This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 15 | 0x0<br>RW/L | **LTR Snoop Requirement Bit Override (LTRSROVR):** This field contains the No-Snoop Requirement bit override value for this particular PCIe root port. This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 14:13 | 0x0<br>RO | **RSVD1:** Reserved |
| 12:10 | 0x0<br>RW/L | **LTR Snoop Latency Scale Override Value (LTRSLSOVRV):** This field contains the Snoop latency scale override value for this particular PCIe root port.<br>• 000: LTRSLOVRV times 1 ns<br>• 001: LTRSLOVRV times 32 ns<br>• 010: LTRSLOVRV times 1,024 ns<br>• 011: LTRSLOVRV times 32,768 ns<br>• 100: LTRSLOVRV times 1,048,576 ns<br>• 101: LTRSLOVRV times 33,554,432 ns<br>• Others: Not Permitted.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 9:0 | 0x0<br>RW/L | **LTR Snoop Latency Override Value (LTRSLOVRV):** This field contains the Snoop latency override value for this particular PCIe root port. This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |

## 47.2.58 INTXSWZCTL—Offset F8h

This register provides software the ability to swizzle the legacy interrupts (INTx) from each port and remap them to a different interrupt type for purposes of interrupt rebalancing. This swizzling only applies to inbound INTx messages that arrive on the secondary side of the switch or root port. This register does not affect PCI-Express defined INTx re-map based on device # that is required by the PCI-Express base specification. This register should be treated as reserved if not applicable to device/ project specific.

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:6, F:0] +F8h
**Offset:** [B:0, D:22, F:0] +F8h
**Offset:** [B:0, D:23, F:0] +F8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RW/L | **INTx Swizzle (INTXSWZ):** The encoding below defines the target INTx type to which the incoming INTx message is mapped to for that port.<br>• 000b:INTA=)INTA, INTB=)INTB, INTC=)INTC, INTD=)INTD<br>• 001b:INTA=)INTB, INTB=)INTC, INTC=)INTD, INTD=)INTA<br>• 010b:INTA=)INTC, INTB=)INTD, INTC=)INTA, INTD=)INTB<br>• 011b:INTA=)INTD, INTB=)INTA, INTC=)INTB, INTD=)INTC<br>• 100b:INTA=)INTA, INTB=)INTA, INTC=)INTA, INTD=)INTA<br><br>**Power Well:** PRST |

## 47.2.59 AERCAPCTL—Offset 118h

This register gives the status and control for ECRC checks and also the pointer to the first uncorrectable error that happened.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +118h
**Offset:** [B:0, D:22, F:0] +118h
**Offset:** [B:0, D:23, F:0] +118h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:13 | 0x0<br>RO | **RSVD0:** Reserved |
| 12 | 0x0<br>RO | **Completion Timeout Prefix/Header Log Capable (CTPHLC):** If Set, this bit indicates that the Function records the prefix/header of Request TLPs that experience a Completion Timeout error. |
| 11 | 0x0<br>RO/V/P | **TLP Prefix Log Present (TLPPLP):** If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information. If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. |
| 10 | 0x0<br>RO | **Multiple Header Recording Enable (MHRE):** When set, this bit enables the function to record more than one error header. |
| 9 | 0x0<br>RO | **Multiple Header Recording Capable (MHRC):** Supports more than one error header. |
| 8 | 0x0<br>RW/P | **ECRC Check Enable (ECE):** When Set, ECRC checking is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. |
| 7 | 0x0<br>RW/L | **ECRC Check Capable (ECC):** If Set, this bit indicates that the Function is capable of checking ECRC.<br><br>**Power Well:** PRST |
| 6 | 0x0<br>RW/P | **ECRC Generation Enable (EGE):** When Set, ECRC generation is enabled. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. Default value of this bit is 0b. |
| 5 | 0x0<br>RW/L | **ECRC Generation Capable (EGC):** If Set, this bit indicates that the Function is capable of generating ECRC.<br><br>**Power Well:** PRST |
| 4:0 | 0x0<br>RO/V/P | **First Error Pointer (FEP):** This field identifies the bit position of the first error reported in the Uncorrectable Error Status Register (xref). This register re-arms itself (which does not change its current value) as soon as the error status bit indicated by the pointer is cleared by the software by writing a 1 to that status bit. |

## 47.2.60 AERHDRLOG1—Offset 11Ch

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +11Ch
**Offset:** [B:0, D:22, F:0] +11Ch
**Offset:** [B:0, D:23, F:0] +11Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 47.2.61 AERHDRLOG2—Offset 120h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +120h
**Offset:** [B:0, D:22, F:0] +120h
**Offset:** [B:0, D:23, F:0] +120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 47.2.62 AERHDRLOG3—Offset 124h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +124h
**Offset:** [B:0, D:22, F:0] +124h
**Offset:** [B:0, D:23, F:0] +124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 47.2.63 AERHDRLOG4—Offset 128h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +128h
**Offset:** [B:0, D:22, F:0] +128h
**Offset:** [B:0, D:23, F:0] +128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 47.2.64 ROOTERRCMD—Offset 12Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +12Ch
**Offset:** [B:0, D:22, F:0] +12Ch
**Offset:** [B:0, D:23, F:0] +12Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **Fatal Error Reporting Enable (FERE):** When set, this bit enables the generation of an interrupt when a Fatal error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 1 | 0x0<br>RW | **Non-Fatal Error Reporting Enable (NFERE):** When set, this bit enables the generation of an interrupt when a Non-Fatal error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 0 | 0x0<br>RW | **Correctable Error Reporting Enable (CERE):** When set, this bit enables the generation of an interrupt when a Correctable error is reported by any of the functions in the hierarchy associated with this Root Port. |

## 47.2.65 ROOTERRSTS—Offset 130h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +130h
**Offset:** [B:0, D:22, F:0] +130h
**Offset:** [B:0, D:23, F:0] +130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO/V | **Advanced Error Interrupt Message Number (AEMN):** This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability. For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated. Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register. For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message. The entry must be one of the first 32 entries even if the Function implements more than 32 entries. For a given MSI-X implementation, the entry must remain constant. For remaining fields info, please refer to PCIe Spec. for more details. |
| 26:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW/1C/V/P | **Fatal Error Message Received (FEMR):** Reserved. |
| 5 | 0x0<br>RW/1C/V/P | **Non-Fatal Error message Received (NFEMR):** Reserved. |
| 4 | 0x0<br>RW/1C/V/P | **First Uncorrectable Fatal (FUF):** Reserved. |
| 3 | 0x0<br>RW/1C/V/P | **Multiple Error Fatal/Non-Fatal Received (MEFR):** Reserved. |
| 2 | 0x0<br>RW/1C/V/P | **Error Fatal/Non-Fatal Received (EFR):** Reserved. |
| 1 | 0x0<br>RW/1C/V/P | **Multiple Error Correctable Error Received (MCER):** Reserved. |
| 0 | 0x0<br>RW/1C/V/P | **Correctable Error Received (CER):** Reserved. |

## 47.2.66 ERRSRCID—Offset 134h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +134h
**Offset:** [B:0, D:22, F:0] +134h
**Offset:** [B:0, D:23, F:0] +134h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO/V/P | **Fatal/Non-Fatal Error Source ID (EFSID):** Requester ID of the source when an Fatal or No Fatal Error is received and the Fatal Error Message Received (FEMR) or Non-Fatal Error message Received (NFEMR) bit is not already set. |
| 15:0 | 0x0<br>RO/V/P | **Correctable Error Source ID (ECSID):** Requester ID of the source when a correctable error is received and the Correctable Error Received (CER) bit is not already set. |

## 47.2.67 ACSCAPHDR—Offset 138h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +138h
**Offset:** [B:0, D:22, F:0] +138h
**Offset:** [B:0, D:23, F:0] +138h

**Default:** 0001000Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities. Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0xd<br>RO | **Extended Capability ID (ECID):** Identifies the function as Access Control Services capable. |

## 47.2.68 ACSCAP—Offset 13Ch

This register identifies the Access Control Services (ACS) capabilities.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +13Ch
**Offset:** [B:0, D:22, F:0] +13Ch
**Offset:** [B:0, D:23, F:0] +13Ch

**Default:** 005Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Egress Control Vector Size (ECVS):** Indicates the number of bits in the Egress Control Vector. This is set to 00h as ACS P2P Egress Control (ACSP2PEC) bit 5 in this register is 0b. |
| 7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x1<br>RO | **ACS Direct Translated P2P (TACSDTP2P):** Indicates that the component does implement ACS Direct Translated P2P. Required for Root Ports that support Address Translation Services (ATS) and also support peer-to-peer traffic with other Root Ports, required for Switch Downstream Ports, |
| 5 | 0x0<br>RO | **ACS P2P Egress Control (EACSP2PEC):** Hardwired to 0. Indicates that the component does not implement ACS P2P Egress Control. |
| 4 | 0x1<br>RO | **ACS Upstream Forwarding (UACSUF):** Indicates that the component implements ACS Upstream Forwarding. |
| 3 | 0x1<br>RO | **ACS P2P Completion Redirect (CACSP2PRR):** Indicates that the component implements ACS P2P Completion Redirect. This includes Read Completions, AtomicOp Completions, and other Completions, either with or without data. |
| 2 | 0x1<br>RO | **ACS P2P Request Redirect (RACSP2PRR):** Indicates that the component implements ACS P2P Request Redirect. |
| 1 | 0x1<br>RO | **ACS Translation Blocking (BACSTB):** Indicates that the component implements ACS Translation Blocking. |
| 0 | 0x1<br>RO | **ACS Source Validation (VACSSV):** Indicates that the component implements ACS Source Validation. |

## 47.2.69 ACSCTL—Offset 13Eh

This register identifies the Access Control Services (ACS) control bits.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +13Eh
**Offset:** [B:0, D:22, F:0] +13Eh
**Offset:** [B:0, D:23, F:0] +13Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW | **ACS Direct Translated P2P Enable (TACSP2PECE):** When Set, overrides the ACS P2P Request Redirect and ACS P2P Egress Control mechanisms with peer-to-peer Memory Requests whose Address Translation (AT) field indicates a Translated address This bit is ignored if ACS Translation Blocking (B) is enabled. |
| 5 | 0x0<br>RO | **ACS P2P Egress Control Enable (EACSP2PECE):** This is hardwired to 0b as the component does not implement ACS P2P Egress Control. |
| 4 | 0x0<br>RW | **ACS Upstream Forwarding Enable (UACSUFE):** When set, the component forwards upstream any Request or Completion TLPs it receives that were redirected upstream by a component lower in the hierarchy. Note that the U bit only applies to upstream TLPs arriving at a Downstream Port, and whose normal routing targets the same Downstream Port. |
| 3 | 0x0<br>RW | **ACS P2P Completion Redirect Enable (CACSP2PCRE):** Determines when the component redirects peer-to-peer Completions upstream, applicable only to Read Completions whose Relaxed Ordering Attribute is clear. |
| 2 | 0x0<br>RW | **ACS P2P Request Redirect Enable (RACSP2PRRE):** This bit determines when the component redirects peer-to-peer Requests upstream. |
| 1 | 0x0<br>RW | **ACS Translation Blocking Enable (BACSTBE):** When set, the component blocks all upstream Memory Requests whose Address Translation (AT) field is not set to the default value. |
| 0 | 0x0<br>RW | **ACS Source Validation Enable (VACSSVE):** When set, the component validates the Bus Number from the Requester ID of upstream Requests against the secondary / subordinate Bus Numbers. |

## 47.2.70 XPTDEF—Offset 2DCh

Lock Key bit is located in the Personality Lock Key Control Register (PLKCTL.TLCL).

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +2DCh
**Offset:** [B:0, D:22, F:0] +2DCh
**Offset:** [B:0, D:23, F:0] +2DCh

**Default:** 00008000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **RSVD0:** Reserved |
| 29 | 0x0<br>RW/P/L | **Internal Sideband Error Message Disable (SBEMSGD):** This bit when set to 1b will disable the generation of the internal sideband message, thus losing some of the error logging capability. It is used to avoid deadlocks on internal sideband bus (if one is discovered). |
| 28:27 | 0x0<br>RO | **RSVD1:** Reserved |
| 26 | 0x0<br>RW/P/L | **Tx Quick Data Credit Return Override (TXQDCRO):** On the IOSF interface, the data for a TLP is required to be sent in back to back clocks after the request is granted. The design is parameterized to either return data credits on each data queue read for the amount of data read, or return all data credits for a TLP when it is granted, assuming that the returned credits cannot be reused before all the data is read out. This bit switches the mode that the data credits are returned. If the design is compiled with the parameter set to return all credits at once, setting this bit will cause each data queue read to return the credits for the amount of data read from the queue (Tx). |
| 25 | 0x0<br>RW/P/L | **Rx Quick Data Credit Return Override (RXQDCRO):** On the IOSF interface, the data for a TLP is required to be sent in back to back clocks after the request is granted. The design is parameterized to either return data credits on each data queue read for the amount of data read, or return all data credits for a TLP when it is granted, assuming that the returned credits cannot be reused before all the data is read out. This bit switches the mode that the data credits are returned. If the design is compiled with the parameter set to return all credits at once, setting this bit will cause each data queue read to return the credits for the amount of data read from the queue (Rx). |
| 24 | 0x0<br>RO | **RSVD2:** Reserved |
| 23:21 | 0x0<br>RW/P/L | **IDF Master Request Throttle Flow (IDFMRTFCCSEL):** This field selects which traffic (flow) class to throttle. Setting this bit causes the IDF Master to only issue a single request at a time (non-pipeline mode) regardless of the fabric advertised credits.<br>These bits are independent of each other.<br>• xx1: Posted<br>• x1x: Non-Posted<br>• 1xx: Completion<br>• 111: Any traffic class.<br>***Note:*** '111' would limit all traffic (flow) classes to a single credit, i.e. each flow class would have a single request credit without sharing a single credit across all flow classes. |
| 20 | 0x0<br>RO | **RSVD3:** Reserved |
| 19:17 | 0x0<br>RW/P/L | **IOSF Master Request Throttle Flow Control Class Select (IMRTFCCSEL):** This field selects which flow control class to throttle. Setting this bit causes the IOSF(CPT/PPF/SWF) Master to only issue a single request at a time (non-pipeline mode) regardless advertised credits by fabric.<br>• These bits are independent of each other.<br>• xx1: Posted<br>• x1x: Non-Posted<br>• 1xx: Completion<br>• 111: Any flow control class<br>***Note:*** '111' would limit all flow classes to a single credit, i.e. each flow class would have a single request credit without sharing a single credit across all flow classes. |
| 16 | 0x0<br>RO | **RSVD4:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x1<br>RW/P/L | **Disable Inbound Peer to Peer Configuration (DIS_INB_P2PCFG):** This bit is actually a disable of type1 configuration upstream. This bit is used for controlling configuration request accesses through the PCI-Express port (security feature) for Type1 Configuration access upstream direction, Type0 will still get UR'd.<br>• 1: Upstream type1 configuration accesses will be Master Aborted by IP.<br>• 0: Allows Upstream type1 configuration access. This register bit is of type 'write once' and is controlled by BIOS/special initialization firmware and applies only to upstream type1 configuration. The default value is to disable upstream type1 configuration access. |
| 14 | 0x0<br>RW/P/L | **Config Type 1 Check Disable (CFGT1CD):** This bit controls the Express Port's type1 configuration check in both direction. This bit applies only if DIS_INB_P2PCFG is clear. Setting this bit bypasses all type1 configuration checks and forwards the type1 configuration instead of UR. This bit does not apply type1 configuration targeting the IP, which follows the proper check. This is not applicable to NTB. |
| 13 | 0x0<br>RW/P/L | **MSG Route by ID Disable (MSGRIDCD):** This bit controls the Express Port's message Route by ID check in both direction. Setting this bit bypasses messages route by ID checks and forwards the message instead of UR/Drop. This bit does not apply messages targeting the IP, which follows the proper check. This is not applicable to NTB. |
| 12 | 0x0<br>RW/P/L | **Completion Check Disable (CCD):** This bit controls the Express Port's completion check (Requester ID/Transaction ID) in both direction. Setting this bit bypasses all completion checks based and forwards the completion instead of unexpected completion check. This bit does not apply completion targeting the IP, which follows the proper check. This is not applicable to NTB and Root Ports. |
| 11 | 0x0<br>RW/P/L | **ISA Check Disable (ISACD):** This bit controls the Express Port's ISA IO region check in both direction. Setting this bit bypasses ISA IO decode checks and forwards access instead of UR. ISA region check (ISACD) is don't care when bit 9 (IOCD) is set. This is not applicable to NTB. |
| 10 | 0x0<br>RW/P/L | **VGA Check Disable (VGACD):** This bit controls the Express Port's VGA Memory and IO check in upstream direction only. Setting this bit bypasses upstream VGA Memory and IO decode checks and forwards the VGA access upstream instead of UR. VGA Memory checks are don't care when bit 8 (MCD) is set, while VGA IO check is a don't care when bit 9 (IOCD) is set. This is not applicable to NTB. |
| 9 | 0x0<br>RW/P/L | **IO Check Disable (IOCD):** This bit controls the Express Port's IO check in both direction. Setting this bit bypasses all IO decode checks based on base/limit or BAR registers and forwards the IO access. This bit does not apply for any IO region decoded for IP's internal resources. IO txn ) 64K will be UR'd when this bit is set. This is not applicable to NTB. |
| 8 | 0x0<br>RW/P/L | **Memory Check Disable (MCD):** This bit controls the Express Port's Memory check in both direction. Setting this bit bypasses all memory decode checks based on base/limit or BAR registers and forwards the memory access. This bit does not apply for any memory region decoded for IP's internal resources. This is not applicable to NTB. |
| 7 | 0x0<br>RW/P/L | **Reserved ECO 7 (RECO7):** This field is reserved for ECO purpose. |
| 6:5 | 0x0<br>RW/P/L | **Store Forward Override (SFO):** These bits are used to override the store forward of TLPs through the transmit queue. One bit enables the override and the other bit selects whether to force store-forwarding on or off. Note that store-forward on is safer in all cases, but impacts latency. If they clock frequencies are known such that store forwarding is not needed, it can be overridden here.<br>• 0x - No override.<br>• 10 - Force store forward disable.<br>• 11 - Force store forward enable. |
| 4:0 | 0x0<br>RW/P/L | **Reserved ECO 0 (RECO0):** This field is reserved for ECO purpose. |

## 47.2.71 XPTDEF2—Offset 2E0h

Lock Key bit is located in the Personality Lock Key Control Register (PLKCTL.TLCL).

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:6, F:0] +2E0h
**Offset:** [B:0, D:22, F:0] +2E0h
**Offset:** [B:0, D:23, F:0] +2E0h

**Default:** 00001D00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **MSI Address Violation Status (MSIAVS):** MSI address violation check status that does not target 0xFEEx_xxxx while MSIAVCD is not being set. This is only valid for RPs usage. |
| 30:19 | 0x0<br>RW/P/L | **Reserved ECO (RECO):** This field is reserved for ECO purpose. |
| 18 | 0x0<br>RW/P/L | **Hold Retrain during Surprise Removal Disable (HRSRD):** For Surprise Removal, EIP RPs will not allow retrain to occur until the SLOTSTS.PDCS bit has been cleared by software for the removal event. This bit is to disable the above behavior and allow the retrain to occur. This field is only applicable to RPs and should be treated as reserved for other devices. |
| 17 | 0x0<br>RW/P/L | **ECRC Malformed Error Precedence Disable (ECRCMFEPD):** This bit changes the error precedence if a TLP has both a ECRC error and is malform.<br>• When this bit is set to 1, the packet will be logged as an ECRC error.<br>• When set to 0, the packet will be logged as a malform TLP, which is compliant to the PCIe spec errata.<br>Only the type of error logged is affected. If the TLP is malform, it will be dropped with no credit return. If there is only an ECRC error detected on a packet, it will be dropped and credits will be returned per PCI-SIG Errata. |
| 16 | 0x0<br>RW/P/L | **MSI Address Violation Check Disable (MSIAVCD):** Disable MSI address violation check that does not target 0xFEEx_xxxx. This is only valid for RPs usage. |
| 15 | 0x0<br>RW/P/L | **Force Downstream Relaxed Order (FORCERODN):** Forces all downstream completions to follow relaxed ordering rules. This allows downstream completions to pass downstream posted requests. Note that setting this bit breaks the producer consumer model when the data is in the address space below the port and the flag is in the address space outside the port, because the read of the flag will not push the data writes down the port. The belief is that usage model is not used. |
| 14 | 0x0<br>RW/P/L | **Downstream Relaxed Order Disable (DNRODIS):** Disables the ability of downstream relaxed ordered completions from passing posted requests in the same direction. |
| 13 | 0x0<br>RW/P/L | **Upstream Relaxed Order Disable (UPRODIS):** Disables the ability of upstream relaxed ordered completions from passing posted requests in the same direction. |
| 12 | 0x1<br>RW/P/L | **IOSF tdparity check disable (TDPAR_CHK_DIS):** When set this bit disables the data parity check for incoming data into the IOSF Target block. The logic will ignore the data parity bit and will operate as though the parity bit did not exist. No errors will be logged and good DW based parity will be written into the data queue. |
| 11 | 0x1<br>RW/P/L | **IOSF tcparity check disable (TCPAR_CHK_DIS):** When set this bit disables the command parity check for incoming commands into the IOSF Target block. The logic will ignore the command parity bit and will operate as though the parity bit did not exist. No errors will be logged and good DW based parity will be written into the header queue. |
| 10 | 0x1<br>RW/P/L | **MCTP Message Bus Number Check Enable (MMBNCE):** When set, if the Requester ID bus number of an MCTP message request received on the PCIe link does not fall within the range between SCBN and SBBN, the MCTP message will be silently discarded. |
| 9 | 0x0<br>RW/P/L | **Completion credit advertisement enable (CPLCDT_EN):** Endpoints are required to advertise infinite completion credits. Setting this defeature bit breaks that rule and enables the advertisement of non-infinite completion credits and flow control updates for completion credits. This is only valid for project specific |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8 | 0x1<br>RW/P/L | **Enable Virtual Port IDs (EVPID):**<br>• 1: Enables SB Virtual Port ID mode with allocated VPIDs which is associated with ports for all the egress INTx and LTR messages based on project specific. We will assign (N-1) virtual sideband PortID's to associate with the physical PortID which is port 0. The physical one is the same one as being assigned by the router as we have now, and would be used for all non-INTx messages and INTx messages from port 0. However, the VPIDs would be used for INTx messages routing from port(N-1:1), respectively. This allows all INTx from N ports to be forwarded to the ILB accordingly, so ILB can differentiate the source of the INTx message.<br>• 0: Normal operation with single SB Port ID per cluster for INTx delivery. This is only valid for project specific with multi root ports. |
| 7 | 0x0<br>RW/P/L | **Enable Receive Completion Ordering (ERCO):** The default ordering is to allow received completions to pass received posted requests to avoid receiver overflow conditions if posted requests cannot be processed. Setting this bit will enable PCI Express ordering for completions, preventing completions from passing posted requests. This is only valid for project specific |
| 6 | 0x0<br>RW/P/L | **Enable MSI Generation Override (MSIGENOV):**<br>• 1: Enables Root Port MSI generation override even if PCICMD.BME = 0 for the root port device as required by PCI Express specification.<br>• 0: Normal operation for Root Port behavior. If PCICMD.BME = 0, the root port will not generate a MSI. This is only valid for project specific . |
| 5 | 0x0<br>RW/P/L | **Completion Combining Disable (CPLCOMBINE_D):** When a larger read request is split into smaller read requests, the completions for the small requests may be opportunistically combined into larger completions to enable higher bandwidth and link utilization, as long as packet formation rules are not violated. This bit disables the port from combining completions. This is only valid for project specific . |
| 4 | 0x0<br>RW/P/L | **Config Retry Disable (CRD):** Setting this bit disables the re-issuing of a configuration request if the request is completed with a configuration retry status (CRS). The completion with CRS will be returned to the requestor. |
| 3 | 0x0<br>RW/P/L | **Optional Unexpected Completion Check Disable (UCCHKD):** Setting this bit disables some optional unexpected checks on completions. Specifically, it disables the checking on the byte count and lower address fields. |
| 2 | 0x0<br>RW/P/L | **MCTP Disable Bus Check (MCTPDBC):** Setting this bit will disable the Bus 0 check for all MCTP VDM Type1 messages. The check is enabled by default as following:<br>• Upstream - ReqBus =/= 0<br>• Downstream - TgtBus =/= 0 |
| 1 | 0x0<br>RW/P/L | **Reserved ECO 1 (RECO1):** This field is reserved for ECO purpose. |
| 0 | 0x0<br>RW/P/L | **Global Stop Disable (GBLSTOPD):** When the design detects an uncorrectable internal error, a stop signal in the design asserts, blocking the processing of transactions in both directions in the transaction layer. Setting this bit, disables the stop mechanism, allowing the transactions to proceed as if no error occurred. |

## 47.2.72 IOSFDEVCLKGCTL—Offset 2E4h

It is included for DMI in RC as an extension of Rootport definition.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +2E4h
**Offset:** [B:0, D:22, F:0] +2E4h
**Offset:** [B:0, D:23, F:0] +2E4h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block controls the delay before going into IDLE. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 47.2.73 SBDEVCLKGCTL—Offset 2E6h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +2E6h
**Offset:** [B:0, D:22, F:0] +2E6h
**Offset:** [B:0, D:23, F:0] +2E6h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x1<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 47.2.74 IDFDEVCLKGCTL—Offset 2E8h

The IDF Device Clock Gate control register is defined only for the Virtual Root port/ Switch Port.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:6, F:0] +2E8h
**Offset:** [B:0, D:22, F:0] +2E8h
**Offset:** [B:0, D:23, F:0] +2E8h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

§ §

# 48 PCI Express Root Ports - B0, D(9-12, 14-17), F0

## 48.1 Introduction and Index

The host-accessible registers for the PCI Express Root Ports are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 48.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device (9-12, 14-17) (decimal), Function 0. The offset addresses are listed.

**Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 1 of 7)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "VID—Offset 0h" on page 1600 |
| 2 | 2 | 19A4 | "DID—Offset 2h" on page 1600 |
| 4 | 2 | 0000 | "PCICMD—Offset 4h" on page 1601 |
| 6 | 2 | 0010 | "PCISTS—Offset 6h" on page 1602 |
| 8 | 1 | 00 | "RID—Offset 8h" on page 1603 |
| 9 | 3 | 060400 | "CCR—Offset 9h" on page 1603 |
| C | 1 | 00 | "CLS—Offset Ch" on page 1604 |
| D | 1 | 00 | "PLAT—Offset Dh" on page 1604 |
| E | 1 | 01 | "HDR—Offset Eh" on page 1604 |
| F | 1 | 00 | "BIST—Offset Fh" on page 1605 |
| 18 | 1 | 00 | "PRIBUS—Offset 18h" on page 1605 |
| 19 | 1 | 00 | "SECBUS—Offset 19h" on page 1606 |
| 1A | 2 | 00 | "SUBBUS—Offset 1Ah" on page 1606 |
| 1C | 1 | 00 | "IOBASE—Offset 1Ch" on page 1607 |
| 1D | 1 | 00 | "IOLIMIT—Offset 1Dh" on page 1607 |
| 1E | 2 | 0000 | "SECSTS—Offset 1Eh" on page 1608 |
| 20 | 2 | 0000 | "MEMBASE—Offset 20h" on page 1609 |
| 22 | 2 | 0000 | "MEMLIMIT—Offset 22h" on page 1610 |
| 24 | 2 | 0001 | "PFBASE—Offset 24h" on page 1611 |
| 26 | 2 | 0001 | "PFLIMIT—Offset 26h" on page 1612 |
| 28 | 4 | 00000000 | "PFBASEU—Offset 28h" on page 1612 |
| 2C | 4 | 00000000 | "PFLIMITU—Offset 2Ch" on page 1613 |
| 34 | 1 | 40 | "CAPPTR—Offset 34h" on page 1613 |
| 3C | 1 | 00 | "INTL—Offset 3Ch" on page 1614 |
| 3D | 1 | 01 | "INTP—Offset 3Dh" on page 1614 |
| 3E | 2 | 0000 | "BCTL—Offset 3Eh" on page 1615 |
| 40 | 2 | 8010 | "EXPCAPLST—Offset 40h" on page 1617 |
| 42 | 2 | 0042 | "EXPCAPLST—Offset 40h" on page 1617 |
| 44 | 4 | 00008021 | "DEVCAP—Offset 44h" on page 1618 |
| 48 | 2 | 2000 | "DEVCTL—Offset 48h" on page 1619 |
| 4A | 2 | 0000 | "DEVSTS—Offset 4Ah" on page 1620 |
| 4C | 4 | 09794883 | "LINKCAP—Offset 4Ch" on page 1621 |
| 50 | 2 | 0000 | "LINKCTL—Offset 50h" on page 1623 |
| 52 | 2 | 1001 | "LINKSTS—Offset 52h" on page 1625 |
| 54 | 4 | 00000000 | "SLOTCAP—Offset 54h" on page 1626 |
| 58 | 2 | 03C0 | "SLOTCTL—Offset 58h" on page 1628 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 2 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 5A | 2 | 0040 | "SLOTSTS—Offset 5Ah" on page 1630 |
| 5C | 2 | 0000 | "ROOTCTL—Offset 5Ch" on page 1632 |
| 5E | 2 | 0001 | "ROOTCAP—Offset 5Eh" on page 1633 |
| 60 | 4 | 00000000 | "ROOTSTS—Offset 60h" on page 1633 |
| 64 | 4 | 00000437 | "DEVCAP2—Offset 64h" on page 1634 |
| 68 | 2 | 0000 | "DEVCTL2—Offset 68h" on page 1637 |
| 6C | 4 | 0000000E | "LINKCAP2—Offset 6Ch" on page 1639 |
| 70 | 2 | 0003 | "LINKCTL2—Offset 70h" on page 1640 |
| 72 | 2 | 0000 | "LINKSTS2—Offset 72h" on page 1642 |
| 80 | 2 | 8801 | "PMCAPLST—Offset 80h" on page 1642 |
| 82 | 2 | C803 | "PMCAPLST—Offset 80h" on page 1642 |
| 84 | 2 | 0008 | "PMCSR—Offset 84h" on page 1644 |
| 86 | 2 | 00 | "PMBSE—Offset 86h" on page 1644 |
| 88 | 2 | 900D | "SSCAPLST—Offset 88h" on page 1645 |
| 8C | 2 | 8086 | "SSVID—Offset 8Ch" on page 1646 |
| 8E | 2 | 0000 | "SSID—Offset 8Eh" on page 1646 |
| 90 | 2 | 0005 | "MSICAPLST—Offset 90h" on page 1646 |
| 92 | 2 | 0100 | "MSICTL—Offset 92h" on page 1647 |
| 94 | 4 | 00000000 | "MSIADDR—Offset 94h" on page 1647 |
| 98 | 4 | 00000000 | "MSIDATA—Offset 98h" on page 1648 |
| 9C | 4 | 00000000 | "MSIMSK—Offset 9Ch" on page 1649 |
| A0 | 4 | 00000000 | "MSIPENDING—Offset A0h" on page 1649 |
| A8 | 4 | 00000020 | "RPPMCSR—Offset A8h" on page 1650 |
| B0 | 2 | 0000 | "PMCSRS—Offset B0h" on page 1652 |
| C0 | 4 | 00000000 | "PPBIFCTL—Offset C0h" on page 1652 |
| C8 | 4 | 00000000 | "SMICSR—Offset C8h" on page 1653 |
| CC | 4 | 00000000 | "SCICSR—Offset CCh" on page 1654 |
| D0 | 2 | 0000 | "SSCTL—Offset D0h" on page 1655 |
| D2 | 2 | 0000 | "EXPPTMBARCTL—Offset D2h" on page 1656 |
| D4 | 4 | 00000000 | "PPD0—Offset D4h" on page 1657 |
| D8 | 4 | 00000000 | "PBTXNCTL—Offset D8h" on page 1659 |
| E0 | 4 | 00000000 | "PBPHYCTL—Offset E0h" on page 1660 |
| EA | 2 | 0000 | "PLKCTL—Offset EAh" on page 1660 |
| EC | 4 | 00000000 | "LTRCSR—Offset ECh" on page 1661 |
| F0 | 4 | 00000000 | "LTRL—Offset F0h" on page 1662 |
| F4 | 4 | 00000000 | "LTROVR—Offset F4h" on page 1663 |
| F8 | 1 | 00 | "INTXSWZCTL—Offset F8h" on page 1664 |
| FC | 4 | 00000000 | "CFGAGTERR—Offset FCh" on page 1665 |
| 100 | 4 | 13810001 | "AERCAPHDR—Offset 100h" on page 1666 |
| 104 | 4 | 00000000 | "ERRUNCSTS—Offset 104h" on page 1667 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 3 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 108 | 4 | 04400000 | "ERRUNCMSK—Offset 108h" on page 1669 |
| 10C | 4 | 00462030 | "ERRUNCSEV—Offset 10Ch" on page 1671 |
| 110 | 4 | 00000000 | "ERRCORSTS—Offset 110h" on page 1673 |
| 114 | 4 | 0000E000 | "ERRCORMSK—Offset 114h" on page 1674 |
| 118 | 4 | 00001000 | "AERCAPCTL—Offset 118h" on page 1675 |
| 11C | 4 | 00000000 | "AERHDRLOG1—Offset 11Ch" on page 1676 |
| 120 | 4 | 00000000 | "AERHDRLOG2—Offset 120h" on page 1676 |
| 124 | 4 | 00000000 | "AERHDRLOG3—Offset 124h" on page 1676 |
| 128 | 4 | 00000000 | "AERHDRLOG4—Offset 128h" on page 1677 |
| 12C | 4 | 00000000 | "ROOTERRCMD—Offset 12Ch" on page 1677 |
| 130 | 4 | 00000000 | "ROOTERRSTS—Offset 130h" on page 1678 |
| 134 | 4 | 00000000 | "ERRSRCID—Offset 134h" on page 1679 |
| 138 | 4 | 1501000D | "ACSCAPHDR—Offset 138h" on page 1679 |
| 13C | 2 | 005F | "ACSCAPHDR—Offset 138h" on page 1679 |
| 13E | 2 | 0000 | "ACSCTL—Offset 13Eh" on page 1681 |
| 140 | 4 | 04000000 | "ERRUNCDETMSK—Offset 140h" on page 1682 |
| 144 | 4 | 00000000 | "ERRCORDETMSK—Offset 144h" on page 1683 |
| 148 | 4 | 00000000 | "ROOTERRDETMSK—Offset 148h" on page 1683 |
| 150 | 4 | 18010012 | "MCSTCAPHDR—Offset 150h" on page 1684 |
| 154 | 2 | 000F | "MCSTCAPHDR—Offset 150h" on page 1684 |
| 156 | 2 | 0000 | "MCSTCTL—Offset 156h" on page 1684 |
| 158 | 4 | 0000000C | "MCSTBAR—Offset 158h" on page 1685 |
| 15C | 4 | 00000000 | "MCSTUBAR—Offset 15Ch" on page 1685 |
| 160 | 4 | 00000000 | "MCSTRCV—Offset 160h" on page 1685 |
| 168 | 4 | 00000000 | "MCSTBLKALL—Offset 168h" on page 1686 |
| 178 | 4 | 00000000 | "MCSTOLBAR—Offset 178h" on page 1686 |
| 17C | 4 | 00000000 | "MCSTUOLBAR—Offset 17Ch" on page 1686 |
| 180 | 4 | 1901000B | "EINJCAPHDR—Offset 180h" on page 1687 |
| 184 | 4 | 00A00003 | "EINJHDR—Offset 184h" on page 1687 |
| 188 | 2 | 0000 | "EINJCTL—Offset 188h" on page 1688 |
| 200 | 4 | 00010019 | "SECEXPCAP—Offset 200h" on page 1689 |
| 204 | 4 | 00000000 | "LINKCTL3—Offset 204h" on page 1690 |
| 208 | 4 | 00000000 | "LANEERRSTS—Offset 208h" on page 1691 |
| 20C | 2 | 0000 | "LANEEQCTL0—Offset 20Ch" on page 1692 |
| 20E | 2 | 0000 | "LANEEQCTL1—Offset 20Eh" on page 1694 |
| 210 | 2 | 0000 | "LANEEQCTL2—Offset 210h" on page 1696 |
| 212 | 2 | 0000 | "LANEEQCTL3—Offset 212h" on page 1698 |
| 214 | 2 | 0000 | "LANEEQCTL4—Offset 214h" on page 1700 |
| 216 | 2 | 0000 | "LANEEQCTL5—Offset 216h" on page 1702 |
| 218 | 2 | 0000 | "LANEEQCTL6—Offset 218h" on page 1704 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 4 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 21A | 2 | 0000 | "LANEEQCTL7—Offset 21Ah" on page 1706 |
| 280 | 4 | 00000000 | "XPPMDL0—Offset 280h" on page 1708 |
| 284 | 4 | 00000000 | "XPPMDL1—Offset 284h" on page 1708 |
| 288 | 4 | FFFFFFFF | "XPPMCL0—Offset 288h" on page 1708 |
| 28C | 4 | FFFFFFFF | "XPPMCL1—Offset 28Ch" on page 1708 |
| 290 | 2 | 0000 | "XPPMDH—Offset 290h" on page 1709 |
| 292 | 2 | 0F0F | "XPPMCH—Offset 292h" on page 1709 |
| 294 | 4 | 00000000 | "XPPMR0—Offset 294h" on page 1710 |
| 298 | 4 | 00000000 | "XPPMR1—Offset 298h" on page 1713 |
| 29C | 4 | 00000000 | "XPPMEVL0—Offset 29Ch" on page 1716 |
| 2A0 | 4 | 00000000 | "XPPMEVL1—Offset 2A0h" on page 1718 |
| 2A4 | 4 | 00000000 | "XPPMEVH0—Offset 2A4h" on page 1720 |
| 2A8 | 4 | 00000000 | "XPPMEVH1—Offset 2A8h" on page 1721 |
| 2AC | 4 | 00000000 | "XPPMER0—Offset 2ACh" on page 1722 |
| 2B0 | 4 | 00000000 | "XPPMER1—Offset 2B0h" on page 1724 |
| 2B4 | 4 | 00000000 | "XPPMDFXMAT0—Offset 2B4h" on page 1726 |
| 2B8 | 4 | 00000000 | "XPPMDFXMAT1—Offset 2B8h" on page 1727 |
| 2BC | 4 | 00000000 | "XPPMDFXMSK0—Offset 2BCh" on page 1728 |
| 2C0 | 4 | 00000000 | "XPPMDFXMSK1—Offset 2C0h" on page 1728 |
| 2C4 | 4 | 00000000 | "XPPERFCON—Offset 2C4h" on page 1729 |
| 2C8 | 1 | 00 | "XPPERFSTAT—Offset 2C8h" on page 1730 |
| 2D0 | 4 | 00000000 | "XPCDTHROTTLEP—Offset 2D0h" on page 1730 |
| 2D4 | 4 | 00000000 | "XPCDTHROTTLEN—Offset 2D4h" on page 1731 |
| 2D8 | 4 | 00000000 | "XPCDTHROTTLEC—Offset 2D8h" on page 1731 |
| 2E4 | 2 | 0010 | "IOSFDEVCLKGCTL—Offset 2E4h" on page 1732 |
| 2E6 | 2 | 8010 | "SBDEVCLKGCTL—Offset 2E6h" on page 1732 |
| 300 | 4 | 0001000B | "VDHDRCAP—Offset 300h" on page 1733 |
| 304 | 4 | 2A800001 | "IBSTHDRCAP—Offset 304h" on page 1733 |
| 310 | 4 | 00000007 | "IBSTENGLKSPDCAP—Offset 310h" on page 1734 |
| 314 | 4 | 00000000 | "IBSTENGLKSPDCTL—Offset 314h" on page 1735 |
| 31C | 4 | 00000000 | "IBSTENGLKTXRXCFG—Offset 31Ch" on page 1736 |
| 320 | 4 | 00000000 | "IBSTERRCRCSTS0—Offset 320h" on page 1736 |
| 324 | 4 | 00000000 | "IBSTERRRCRVSTS0—Offset 324h" on page 1737 |
| 328 | 4 | 00000000 | "IBSTERRCRCSTS1—Offset 328h" on page 1737 |
| 32C | 4 | 00000000 | "IBSTERRRCRVSTS1—Offset 32Ch" on page 1738 |
| 330 | 4 | 00000000 | "IBSTERRCRCSTS2—Offset 330h" on page 1738 |
| 334 | 4 | 00000000 | "IBSTERRRCRVSTS2—Offset 334h" on page 1739 |
| 338 | 4 | 00000000 | "IBSTERRCRCSTS3—Offset 338h" on page 1739 |
| 33C | 4 | 00000000 | "IBSTERRRCRVSTS3—Offset 33Ch" on page 1740 |
| 344 | 4 | 00400000 | "IBSTPHYLCTL0—Offset 344h" on page 1741 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 5 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 348 | 4 | 00400000 | "IBSTPHYLCTL1—Offset 348h" on page 1742 |
| 34C | 4 | 00400000 | "IBSTPHYLCTL2—Offset 34Ch" on page 1743 |
| 350 | 4 | 00400000 | "IBSTPHYLCTL3—Offset 350h" on page 1744 |
| 354 | 4 | 00000000 | "IBSTTXLCTL—Offset 354h" on page 1745 |
| 358 | 4 | 00000000 | "IBSTTDSTS—Offset 358h" on page 1745 |
| 36C | 4 | 00000000 | "IBSTPLLNSTS0—Offset 36Ch" on page 1746 |
| 370 | 4 | FFFFFFFF | "IBSTPRECFG0—Offset 370h" on page 1747 |
| 374 | 4 | FFFFFFFF | "IBSTPOSTCFG0—Offset 374h" on page 1748 |
| 378 | 4 | 00000000 | "IBSTPLLNSTS1—Offset 378h" on page 1749 |
| 37C | 4 | FFFFFFFF | "IBSTPRECFG1—Offset 37Ch" on page 1750 |
| 380 | 4 | FFFFFFFF | "IBSTPOSTCFG1—Offset 380h" on page 1751 |
| 8E0 | 2 | 001F | "TLPCIECLKGCTL—Offset 8E0h" on page 1751 |
| 8E2 | 2 | 0055 | "TLPCIECLKGCCTL—Offset 8E2h" on page 1752 |
| 8E4 | 2 | 0010 | "PT0TXNCLKGCTL—Offset 8E4h" on page 1752 |
| 8E6 | 2 | 0010 | "PT1TXNCLKGCTL—Offset 8E6h" on page 1753 |
| 8E8 | 2 | 0010 | "PT2TXNCLKGCTL—Offset 8E8h" on page 1753 |
| 8EA | 2 | 0010 | "PT3TXNCLKGCTL—Offset 8EAh" on page 1754 |
| 8F8 | 4 | 00001D00 | "XPTDEF2—Offset 8F8h" on page 1755 |
| 948 | 4 | 00000306 | "PMIDLTMR—Offset 948h" on page 1757 |
| 98C | 4 | 00000000 | "LLRBSIZE—Offset 98Ch" on page 1758 |
| 99C | 2 | 0000 | "FRMERRMSK—Offset 99Ch" on page 1759 |
| 99E | 2 | 0000 | "FRMERRSTS—Offset 99Eh" on page 1760 |
| 9E0 | 2 | 001F | "LLPCIECLKGCTL—Offset 9E0h" on page 1760 |
| 9E2 | 2 | 0055 | "LLPCIECLKGCCTL—Offset 9E2h" on page 1761 |
| 9E4 | 2 | 0010 | "PT0LNKCLKGCTL—Offset 9E4h" on page 1761 |
| 9E6 | 2 | 0010 | "PT1LNKCLKGCTL—Offset 9E6h" on page 1762 |
| 9E8 | 2 | 0010 | "PT2LNKCLKGCTL—Offset 9E8h" on page 1762 |
| 9EA | 2 | 0010 | "PT3LNKCLKGCTL—Offset 9EAh" on page 1763 |
| 9EC | 2 | 0010 | "CFGPCIECLKGCTL—Offset 9ECh" on page 1763 |
| 9EE | 2 | 0001 | "CFGPCIECLKGCCTL—Offset 9EEh" on page 1763 |
| A68 | 4 | 00000000 | "LTSSMSMSTS—Offset A68h" on page 1764 |
| A70 | 4 | 00000000 | "LTSSMLNSTS0—Offset A70h" on page 1765 |
| A80 | 4 | 00000000 | "LTSSMSTATELOG0—Offset A80h" on page 1765 |
| A84 | 4 | 00000000 | "LTSSMSTATELOG1—Offset A84h" on page 1765 |
| A88 | 4 | 00000000 | "LTSSMSTATELOG2—Offset A88h" on page 1766 |
| A8C | 4 | 00000000 | "LTSSMSTATELOGCTL—Offset A8Ch" on page 1766 |
| A94 | 2 | 0000 | "LTLNERRCTRL—Offset A94h" on page 1766 |
| A96 | 2 | 0000 | "LTLNERRSTS—Offset A96h" on page 1767 |
| A98 | 4 | 00000000 | "LTLNERRLOG—Offset A98h" on page 1768 |
| AB8 | 4 | 00000000 | "UPCFGCSR—Offset AB8h" on page 1769 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 6 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| AC8 | 4 | 00025A4E | "SOSCTL—Offset AC8h" on page 1770 |
| ACC | 4 | 00000000 | "G3SOSERRSTS—Offset ACCh" on page 1770 |
| AE0 | 2 | 001F | "PLPCIECLKGCTL—Offset AE0h" on page 1770 |
| AE2 | 2 | 0055 | "PLPCIECLKGCCTL—Offset AE2h" on page 1771 |
| AE4 | 2 | 0010 | "PT0PHYCLKGCTL—Offset AE4h" on page 1771 |
| AE6 | 2 | 0010 | "PT1PHYCLKGCTL—Offset AE6h" on page 1772 |
| AE8 | 2 | 0010 | "PT2PHYCLKGCTL—Offset AE8h" on page 1772 |
| AEA | 2 | 0010 | "PT3PHYCLKGCTL—Offset AEAh" on page 1773 |
| B00 | 4 | 36000000 | "LTLGCAP—Offset B00h" on page 1773 |
| B04 | 4 | 00321000 | "LTLGCTRL—Offset B04h" on page 1774 |
| B08 | 4 | 00000001 | "LTLGSTART0—Offset B08h" on page 1775 |
| B0C | 4 | 00000001 | "LTLGSTART1—Offset B0Ch" on page 1776 |
| B10 | 4 | 00000001 | "LTLGSTART2—Offset B10h" on page 1777 |
| B14 | 4 | 00000001 | "LTLGSTART3—Offset B14h" on page 1778 |
| B18 | 4 | 1E000009 | "LTLGSTP0—Offset B18h" on page 1779 |
| B1C | 4 | 1E000009 | "LTLGSTP1—Offset B1Ch" on page 1779 |
| B20 | 4 | 1E000009 | "LTLGSTP2—Offset B20h" on page 1780 |
| B24 | 4 | 1E000009 | "LTLGSTP3—Offset B24h" on page 1780 |
| B28 | 4 | 00000000 | "LTLGTMEV0—Offset B28h" on page 1781 |
| B2C | 4 | 00000000 | "LTLGTMEV1—Offset B2Ch" on page 1782 |
| B30 | 4 | 00000000 | "LTLGTMEV2—Offset B30h" on page 1783 |
| B34 | 4 | 00000000 | "LTLGTMEV3—Offset B34h" on page 1784 |
| B40 | 4 | 00000000 | "LTLGRDCT—Offset B40h" on page 1784 |
| B44 | 4 | FFFFFFFF | "LTLGRES—Offset B44h" on page 1785 |
| B48 | 4 | 00000000 | "LTLGSMEV0—Offset B48h" on page 1785 |
| B4C | 4 | 00000000 | "LTLGSMEV1—Offset B4Ch" on page 1786 |
| B50 | 4 | 00FFFFFF | "NFTSCOMCLK—Offset B50h" on page 1786 |
| B54 | 4 | 00FFFFFF | "NFTSDSCCLK—Offset B54h" on page 1787 |
| B58 | 4 | 2000FFFF | "LEKBERR—Offset B58h" on page 1787 |
| B5C | 4 | FFFFFFFF | "EXPBERR0—Offset B5Ch" on page 1787 |
| B60 | 4 | 02000007 | "EXPBERR1—Offset B60h" on page 1788 |
| B64 | 4 | 00000000 | "LEKBPROERR—Offset B64h" on page 1788 |
| B68 | 4 | 02001821 | "SPEEDCTL—Offset B68h" on page 1789 |
| B6C | 4 | 00000000 | "RCVRDETSTS—Offset B6Ch" on page 1791 |
| B70 | 4 | 00000000 | "RCVRERR—Offset B70h" on page 1791 |
| B74 | 4 | 00000000 | "EBERR—Offset B74h" on page 1792 |
| B78 | 4 | 00000000 | "EBERRMSK—Offset B78h" on page 1792 |
| B7C | 4 | 00000000 | "OBEINJCTL—Offset B7Ch" on page 1793 |
| B80 | 4 | 00000000 | "LTSSMERRSTS0—Offset B80h" on page 1795 |
| B84 | 4 | 00000000 | "LTSSMERRSTS1—Offset B84h" on page 1797 |

## Table 48-1. Summary of PCI Configuration Registers—0/9-12, 14-17/0 (Sheet 7 of 7)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| B8C | 4 | 04904623 | "EQEVALCTL6—Offset B8Ch" on page 1798 |
| B90 | 4 | 000000FF | "EQEVALCTL7—Offset B90h" on page 1799 |
| B9C | 4 | 00000000 | "EQFARFSLF—Offset B9Ch" on page 1800 |
| BA4 | 4 | 03046820 | "FOMINCCTL—Offset BA4h" on page 1801 |
| BA8 | 4 | 0000000C | "PHYRECAL—Offset BA8h" on page 1801 |
| BAC | 4 | 00000100 | "PLLSTBYCTL—Offset BACh" on page 1802 |
| BB0 | 4 | 078A0000 | "PLUTXEQCTL—Offset BB0h" on page 1802 |
| BB4 | 4 | 00000000 | "PHYCGIDLEMSK—Offset BB4h" on page 1803 |
| BB8 | 4 | 64320000 | "EQEVALCTL4—Offset BB8h" on page 1804 |
| BBC | 4 | 3A87F003 | "EQEVALCTL5—Offset BBCh" on page 1805 |
| BC0 | 4 | 00000000 | "LEKBLNERRCNT0—Offset BC0h" on page 1805 |
| BC4 | 4 | 00000000 | "LEKBLNERRCNT1—Offset BC4h" on page 1806 |
| BD0 | 4 | 00080808 | "EQEVALCTL6—Offset B8Ch" on page 1798 |
| BD4 | 4 | 0000FF55 | "EQEVALCTL2—Offset BD4h" on page 1807 |
| BD8 | 4 | 33F30802 | "EQEVALCTL3—Offset BD8h" on page 1808 |
| BDC | 4 | 00000000 | "EQEVALSTS—Offset BDCh" on page 1809 |
| BE0 | 4 | 00000000 | "EQEVALSTS2—Offset BE0h" on page 1811 |
| BE4 | 4 | 00000000 | "EQEVALSTS3—Offset BE4h" on page 1812 |
| BEC | 4 | 00000000 | "G3FRAMEERR—Offset BECh" on page 1814 |

### 48.1.2 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 48-2. Summary of Sideband Registers—0xB4 (Devices 9-12)**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 10 | 00000000 | "PPBIFCTL_PRIV—Offset 10h" on page 1815 |

**Table 48-3. Summary of Sideband Registers—0xB3 (Devices 14-17)**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 10 | 00000000 | "PPBIFCTL_PRIV—Offset 10h" on page 1816 |

## 48.2 Registers in Configuration Space

### 48.2.1 VID—Offset 0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies Intel as the manufacturer of the device. |

### 48.2.2 DID—Offset 2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2h

**Default:** 19A4h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | *See the Field Description for the default value for each Root Port*<br><br>RO/V | **Device ID (DID):** This field identifies the particular function as allocated by Intel.<br>The default ID values assigned to each Host Root Port:<br>• RP0 = 0x19A4 (PCI Device 9)<br>• RP1 = 0x19A5 (PCI Device 10)<br>• RP2 = 0x19A6 (PCI Device 11)<br>• RP3 = 0x19A7 (PCI Device 12)<br>• RP4 = 0x19A8 (PCI Device 14)<br>• RP5 = 0x19A9 (PCI Device 15)<br>• RP6 = 0x19AA (PCI Device 16)<br>• RP7 = 0x19AB (PCI Device 17) |

## 48.2.3 PCICMD—Offset 4h

This register controls how the device behaves on the primary interface (PCI Express).

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (INTxD):** This bit controls the ability of the PCI-Express Function to generate legacy INTx interrupt message.<br>When set, functions are prevented from asserting INTx interrupt messages.<br>Any INTx emulation interrupts already asserted by the function must be deasserted when this bit is set by generating a Deassert_INTx message(s). |
| 9 | 0x0<br>RO | **Fast Back-to-back enable (FBE):** Not applicable to PCI-Express. Hardwired to 0. |
| 8 | 0x0<br>RW | **SERR# Enable (SEE):** When set, this bit enables reporting of Non-Fatal and Fatal errors detected by the Function of the Root Complex. |
| 7 | 0x0<br>RO | **Wait Cycle Control (WCC):** Not applicable to PCI-Express. Hardwired to 0. |
| 6 | 0x0<br>RW | **Parity Error Response Enable (PERE):** This bit controls the setting of the master data parity error bit in the Status Register ('PCI Status Register "PCISTS—Offset 6h" on page 1602) in response to a parity error received on the PCI Express interface (either internal queue errors or a poisoned TLP). |
| 5 | 0x0<br>RO | **VGA Palette Snoop Enable (VGA_PSE):** Not applicable to PCI-Express. Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not applicable to PCI-Express. Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not applicable to PCI-Express. Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** This bit controls the ability of the Function to issue Memory and I/O read or write requests, and the ability of Root or Switch port to forward memory and I/O read or write requests in the upstream direction.<br>When this bit is 0b, memory and I/O requests received at the root port or downstream side of a switch port (secondary side) must be handled as an Unsupported Request (UR). |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):**<br>• 0 - Respond to all Memory Requests on the primary interface as Unsupported Request Received.<br>• 1 - Enable forwarding of memory transactions to the secondary interface and any internal function. |
| 0 | 0x0<br>RW | **I/O Space Enable (IOSE):**<br>• 0 - Respond to all I/O Requests on the primary interface with an Unsupported Request Completion.<br>• 1 - Enable forwarding of I/O Requests to the secondary interface. |

## 48.2.4 PCISTS—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Detected Parity Error (DPE):** This bit is set when a poisoned TLP is received from PCIe or IOSF. This bit is set even when the parity error response enable bit (bit[6] of the PCICMD Register 'PCI Command Register "PCICMD—Offset 4h" on page 1601) is not set. |
| 14 | 0x0<br>RW/1C/V | **Signaled System Error (SSE):** This bit is set when ERR_FATAL or ERR_NONFATAL messages are sent to the root complex and the SERR enable bit in the PCICMD Register ('PCI Command Register "PCICMD—Offset 4h" on page 1601) is set. |
| 13 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit is set when the requester receives a completion with an UR completion status.<br>On Type 1 configuration header functions, the bit is set when a UR completions status is received on the primary side. |
| 12 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit is set when a requester receives a CA completions status.<br>On Type 1 configuration header functions, the bit is set when a 'Completer Abort' is received on the primary side. |
| 11 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit is Set when the port completes a Posted or Non-Posted Request as a Completer Abort error.<br>This applies to a Function with a Type 1 Configuration header when the Completer Abort was generated by its Primary Side. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DVT):** These bits have no meaning on PCI Express. |
| 8 | 0x0<br>RW/1C/V | **Master Data Parity Error Detected (MDPD):** This bit is set by a requester (primary side for type1 configuration header functions) if the parity error response enable bit (PERE) in the Command Register ('PCI Command Register "PCICMD—Offset 4h" on page 1601) is set and either of the following two conditions occur: |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** This bit has no meaning on PCI Express. |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** This bit has no meaning on PCI Express. |
| 4 | 0x1<br>RO | **Capabilities List Enable (CAPE):** This bit indicates the presence of an Extended capabilities list items. Offset 34H indicates the offset for the first entry in the linked list of capabilities. |
| 3 | 0x0<br>RO/V | **Interrupt Status (INTS):** When set, this bit indicates that an INTx emulation interrupt is pending internally in this function. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 48.2.5 RID—Offset 8h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V/P | **Revision ID (RID):** This field specifies the specific revision of this function and should be viewed as a extension to the Device ID. |

## 48.2.6 CCR—Offset 9h

The CCR register has an attribute of RW-L PRST to allow configuration software to modify and lock the register from further update. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9h

**Default:** 060400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x6<br>RW/L | **Base Class (BC):** The value of 06h indicates that this is a bridge device.<br><br>**Power Well:** PRST |
| 15:8 | 0x4<br>RW/L | **Sub-Class (SC):** This 8-bit value indicates that this device is a PCI-to-PCI Bridge.<br><br>**Power Well:** PRST |
| 7:0 | 0x0<br>RW/L | **Register-Level Programming Interface (RLPI):** This bit indicates that this device is standard (non-subtractive) PCI-to-PCI Bridge.<br><br>**Power Well:** PRST |

## 48.2.7 CLS—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Cache Line Size (CLS):** These bits specify the system cache-line size in units of DWords. |

## 48.2.8 PLAT—Offset Dh

This register denotes the maximum time slice for a burst transaction in legacy PCI 2.3 on the primary interface. It does not affect/influence PCI-Express functionality.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Primary Latency Timer (Prim_Lat_timer):** Not applicable to PCI-Express. Hardwired to 00h. This register is typically called as 'latency Timer' for Endpoints in PCI 3.0 specification |

## 48.2.9 HDR—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + Eh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/L | **Multi-function device (MFD):** The root port, DMI in RC, Virtual Root port/switch port and downstream ports are single function only.<br><br>**Power Well:** PRST |
| 6:0 | 0x1<br>RW/L | **Header Type (HTYPE):** These bits define the layout of addresses 10h through 3Fh in the configuration space.<br>These bits read as 01h to indicate that the register layout conforms to the standard PCI-to-PCI Bridge layout.<br><br>**Power Well:** PRST |

## 48.2.10 BIST—Offset Fh

This register is used for reporting control and status information of BIST checks within a PCIExpress port. It is not supported in this implementation.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **BIST Tests (BIST_TST):** Not supported. Hardwired to 0h |

## 48.2.11 PRIBUS—Offset 18h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 18h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Primary Bus Number (PBN):** These bits indicate the PCI Express bus number. |

## 48.2.12 SECBUS—Offset 19h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 19h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Secondary Bus Number (SCBN):** These bits indicate the bus number of the PCI device to which the secondary interface is connected. |

## 48.2.13 SUBBUS—Offset 1Ah

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 1Ah

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Subordinate Bus Number (SBBN):** These bits indicate the highest PCI bus number downstream of this bridge. |

## 48.2.14 IOBASE—Offset 1Ch

The I/O Base and I/O Limit registers define an address range that is used by the PCI-Express port to determine when to forward I/O transactions from one interface to the other using the following formula:

IO_BASE (= A[15:12](=IO_LIMIT Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0.

The bottom of the defined I/O address range will be aligned to a 4KB boundary while the top of the region specified by IO_LIMIT will be one less than a 4 KB multiple.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 1Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW | **I/O Base Address Bits (IOBA):** These bits define the bottom address of an address range to determine when to forward I/O transactions from one interface to another.<br>These bits correspond to address lines[15:12] for 4 KB alignment. Bits[11:0] are assumed to be 000h. |
| 3:0 | 0x0<br>RO | **I/O Base Addressing Capability (IOBC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only. |

## 48.2.15 IOLIMIT—Offset 1Dh

The I/O Base and I/O Limit registers define an address range that is used by the PCI-Express bridge to determine when to forward I/O transactions from one interface to the other using the following formula:

IO_BASE (= A[15:12] (=IO_LIMIT Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] of the IO limit register is treated as FFFh.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 1Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW | **I/O Limit Address Bits (IOLA):** These bits define the top address of an address range to determine when to forward I/O transactions from PCI Express to PCI. These bits correspond to address lines[15:12] for 4 KB aligned window. |
| 3:0 | 0x0<br>RO | **I/O Limit Addressing Capability (IOLC):** Each of these bits is hard-wired to 0, indicating support for 16-bit I/O addressing only. |

## 48.2.16 SECSTS—Offset 1Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 1Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Detected Parity Error (DPE):** This bit is set by the secondary side for a Type 1 Configuration Space header function whenever it receives a Poisoned TLP, regardless of the state in the Parity Error Response Enable (PERE) field of the Bridge Control Register (BCTL). |
| 14 | 0x0<br>RW/1C/V | **Received System Error (RSE):** This bit is set by the secondary side for a Type 1 Configuration Space header function whenever it receives an ERR_FATAL or ERR_NONFATAL message. |
| 13 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit is set when the secondary side for Type 1 configuration space header function (for requests initiated by the Type 1 header function itself) receives a completion with Unsupported Requests Completion Status. |
| 12 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit is set when the secondary side for Type 1 Configuration Space Header Function (for Requests initiated by the Type 1 header Function itself) receives a completion with Completer About Completion Status. |
| 11 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit is set when the secondary side for Type 1 configuration space header function (for requests completed by type 1 header functions itself) completes a Posted or Non-posted request as a Completer Abort error. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DVT):** Not applicable to PCI Express. Hardwired to 0. |
| 8 | 0x0<br>RW/1C/V | **Master Data Parity Error Detected (MDPD):** This bit is set by the secondary side requester if the Parity Error Response Enable (PERE) bit in the Bridge Control Register (BCTL) is set and either of the following conditions occurs:<br>Requester receives completion marked poisoned Requester sends a poisoned request (includes writes and messages) If the PERE bit in the Bridge Control Register (BCTL) is clear, this bit is never set. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not applicable to PCI Express. Hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not applicable to PCI Express. Hardwired to 0. |
| 4:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 48.2.17 MEMBASE—Offset 20h

The Memory Base and Memory Limit registers define a memory mapped I/O non-prefetchable address range (32-bit addresses) and direct the accesses in this range to the PCI-Express port based on the following formula:

MEMORY_BASE (= A[31:20] (= MEMORY_LIMIT The upper 12 bits of both the Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, AD[31:20], of 32-bit addresses.

For the purpose of address decoding, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 20h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Memory Base (MB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range. |
| 3:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 48.2.18 MEMLIMIT—Offset 22h

This register controls the processor to PCI-Express non-prefetchable memory access routing based on the following formula as described above:

MEMORY_BASE (= A[31:20] (= MEMORY_LIMIT The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. The bottom 4 bits of this register are read-only and return zeros when read. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 22h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Memory Limit (ML):** These bits are compared with bits[31:20] of the incoming address to determine the upper limit of the range of memory accesses that will be passed by the port. |
| 3:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 48.2.19 PFBASE—Offset 24h

The Prefetchable Memory Base and Memory Limit registers define a memory mapped I/O prefetchable address range (32-bit addresses) which is used by the PCI-Express bridge to determine when to forward memory transactions based on the following formula:

PREFETCH_MEMORY_BASE (= A[31:20] (= PREFETCH_MEMORY_LIMIT The upper 12 bits of both the Prefetchable Memory Base and Memory Limit registers are read/ write and corresponds to the upper 12 address bits, A[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, A[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, A[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 24h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Prefetchable Memory Base (PMB):** These bits are compared with bits[31:20] of the incoming address to determine the lower 1 MB-aligned value (inclusive) of the range.<br>The incoming address must be greater than or equal to this value. |
| 3:0 | 0x1<br>RO | **Prefetchable Memory Base Address Capability (PFMCAP):**<br>• 0: 32-bit Prefetchable Memory addressing.<br>• 1: 64-bit Prefetchable Memory addressing.<br>This field indicates that 64-bit addressing is supported for the limit. |

## 48.2.20 PFLIMIT—Offset 26h

This register controls the processor to PCI-Express prefetchable memory access routing based on the following formula as described above:

PREFETCH_MEMORY_BASE (= A[31:20] (= PREFTCH_MEMORY_LIMIT The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 26h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RW | **Prefetchable Memory Limit (PML):** These bits are compared with bits[31:20] of the incoming address to determine the upper limit of the prefetchable memory range. |
| 3:0 | 0x1<br>RO | **Prefetchable Memory Limit Address Capability (PMLC):** This field indicates that 64-bit addressing is supported for the limit. |

## 48.2.21 PFBASEU—Offset 28h

The Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers are extensions to the Prefetchable Memory Base and Prefetchable Memory Limit registers. If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers specify the upper 32-bits, corresponding to A[63:32], of the 64-bit base and limit addresses which specify the prefetchable memory address range. PREFETCH_MEMORY_BASE_UPPER (= A[63:32] (= PREFETCH_MEMORY_BASE_LIMIT_UPPER

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Prefetchable Memory Base Upper Portion (PMBU):** Corresponds to A[63:32] of the memory address that maps to the upper base of the prefetchable range of memory accesses that will be passed by the PCI-Express bridge. |

## 48.2.22 PFLIMITU—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Prefetchable Memory Limit Upper Portion (PMLU):** Corresponds to A[63:32] of the memory address that maps to the upper limit of the prefetchable range of memory accesses that will be passed by the PCI-Express bridge. |

## 48.2.23 CAPPTR—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 34h

**Default:** 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CPTR):** Contains the offset of the first item in the list of capabilities. (EXPCAPLST) |

## 48.2.24 INTL—Offset 3Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 3Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information. The device itself does not use this value, rather it is used by device drivers and operating systems. (compatibility) |

## 48.2.25 INTP—Offset 3Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 3Dh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RW/L | **Interrupt Pin (INTP):** This register tells which interrupt pin the function uses.<br>• 01h: Generate INTA<br>• 02h: Generate INTB<br>• 03h: Generate INTC<br>• 04h: Generate INTD<br>• Others: Reserved BIOS has the ability to write this register once during boot to setup the correct interrupt for the Function.<br><br>**Power Well:** PRST |

## 48.2.26 BCTL—Offset 3Eh

This register provides extensions to the Command Register ('PCI Command Register "PCICMD—Offset 4h" on page 1601) that are specific to a bridge. The Bridge Control Register provides many of the same controls for the secondary interface that are provided by the Command Register for the primary interface. Some bits affect operation of both interfaces of the bridge.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 3Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11 | 0x0<br>RO | **Discard Timer SERR# Enable (DTSE):** Not applicable to PCI Express. Hardwired to 0. |
| 10 | 0x0<br>RO | **Discard Timer Status (DTS):** Not applicable to PCI Express. Hardwired to 0. |
| 9 | 0x0<br>RO | **Secondary Discard Timer (SDT):** Not applicable to PCI Express. Hardwired to 0. |
| 8 | 0x0<br>RO | **Primary Discard Timer (PDT):** Not applicable to PCI Express. Hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not applicable to PCI Express. Hardwired to 0. |
| 6 | 0x0<br>RW | **Secondary Bus Reset (SBR):** Setting this bit triggers a hot reset on the downstream link for the corresponding PCIe Express port and the PCI Express hierarchy domain subordinate to the port.<br>Software must ensure a minimum reset duration of 1us as defined in the PCI Local Bus Specification, Revision 3.0.<br>Hardware will continue to maintain the hot reset state as long as the SBR bit is set.<br>For Root Ports, it is recommended that software assert this field for a minimum of 2 ms to ensure that all downstream links enters hot reset state.<br>A secondary bus reset will not reset any register of a Type 1 configuration space header function. |
| 5 | 0x0<br>RO | **Master Abort Mode (MAM):** Not applicable to PCI Express. Hardwired to 0. |
| 4 | 0x0<br>RW | **VGA 16-bit Decode (VGA16BD):** This bit enables the bridge to provide 16-bit decoding of VGA I/O address precluding the decoding of VGA alias addresses every 1 KB.<br>This bit requires the VGA enable bit (bit 3 of this register) to be set to 1.<br>• 0: execute 10-bit address decode on VGA I/O accesses<br>• 1: execute 16-bit address decode on VGA I/O accesses |
| 3 | 0x0<br>RW | **VGA Enable (VGAE):** This bit modifies the response to VGA-compatible addresses.<br>When set to 1b, the bridge positively decodes and forwards the following transactions from primary side to secondary side regardless of the value of the I/O base and limit registers.<br>The transactions are qualified by the memory enable and I/O enable in the command register.<br>Memory addresses:<br>000A 0000h-000B FFFFh I/O addresses: 3B0h-3BBh and 3C0h-3DFh in first 64 KB of I/O address space (Inclusive of ISA address aliases when IO address bits[15:10] are not decoded) |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **ISA Enable (IE):** This bit modifies the response by the bridge to ISA I/O addresses.<br>This field applies only to I/O addresses that are enabled by the I/O base and I/O limit registers and are in the first 64 KB of PCI I/O space.<br>When this bit is set, the bridge blocks all forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1 KB block (offsets 100h to 3FFh).<br>In the opposite direction (secondary to primary), I/O transactions will be forwarded if they address the last 768B in each 1 KB block.<br>• 1: Forward upstream ISA I/O addresses in the address range defined by the I/O Base and I/O Limit registers that are in the firsts 64KB of PCI I/O address space (Top 768B of each 1K block).<br>• 0: Forward downstream all I/O addresses in the address range defined by the I/O Base and I/O Limit registers. |
| 1 | 0x0<br>RW | **SERR# Enable (SE):** This bit controls the forwarding of PCI Express ERR_COR, ERR_NONFATAL and ERR_FATAL messages to the primary side.<br>• 1: Enables forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL messages.<br>• 0: Disables forwarding of ERR_COR, ERR_NONFATAL, ERR_FATAL messages. |
| 0 | 0x0<br>RW | **Parity Error Response Enable (PERE):** This bit controls the response to poisoned TLPs in the PCI Express port.<br>• 1: Enables reporting of poisoned TLP errors.<br>• 0: Disables reporting of poisoned TLP errors. |

## 48.2.27 EXPCAPLST—Offset 40h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 40h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x80<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list. (PMCAPLST) |
| 7:0 | 0x10<br>RO | **Capability ID (CAPID):** Identifies the function as PCI Express capable. |

## 48.2.28 EXPCAP—Offset 42h

This register stores the version number of the capability item and other base information contained in the capability structure.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 42h

**Default:** 0042h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13:9 | 0x0<br>RO/V | **Interrupt Message Number (IMN):** This field indicates the interrupt message number that is generated from the PCI Express port.<br>When there is more than one MSI interrupt number, this register is required to contain the offset between the base Message Data and the MSI Message that is generated when the status bits in the slot status register or root port status registers are set.<br>The chipset is required to update this field if the number of MSI messages change.<br>Applies only to Root ports. The PCIe macro assigns the first vector for PM/HP events and so this field is set to 0. |
| 8 | 0x0<br>RW/L | **Slot Implemented (SI):**<br>• 1: Indicates the PCI Express link associated with this port is connected to a slot.<br>• 0: Indicates no slot is connected to this port. Applies only to root and downstream ports.<br>***Note:*** For any devices with hot plug enabled, this bit will be forced to 1 (slot) in BIOS.<br>**Power Well:** PRST |
| 7:4 | 0x4<br>RO | **Device/Port Type (DT):** 4h: Root Port of a PCIe Root Complex |
| 3:0 | 0x2<br>RO | **Version Number (VN):** These bits indicate the version number of the PCI Express capability structure. |

## 48.2.29 DEVCAP—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 44h

**Default:** 00008021h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:26 | 0x0<br>RO | **Captured Slot Power Limit Scale (CSPLS):** In combination with the Slot Power Limit value (bits[25:18], this field specifies the upper limit of the power supplied by slot.<br>The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Value field.<br>This value is set by the Set_Slot_Power_Limit message.<br>Does not apply to root ports or integrated devices or end points or downstream ports or Virtual Root Port/switch port |
| 25:18 | 0x0<br>RO | **Captured Slot Power Limit Value (CSPLV):** In combination with the Slot Power Limit Scale value (bits[27:26]), this field specifies the upper limit of the power supplied by slot.<br>The power limit (in Watts) is calculated by multiplying the value in this field by the value in the Slot Power Limit Scale field.<br>This value is set by the Set_Slot_Power_Limit message.<br>Does not apply to root ports or integrated devices or end points or downstream ports or Virtual Root Port/switch ports |
| 17:16 | 0x0<br>RO | **RSVD1:** Reserved |
| 15 | 0x1<br>RO | **Role-Based Error Reporting (RBER):** The PCIe cluster supports Role-based Error Reporting. |
| 14:12 | 0x0<br>RO | **Undefined:** Reserved. |
| 11:9 | 0x0<br>RO | **Endpoint L1 Acceptable Latency (EPL1AL):** This field only applies to Endpoint and is hardwired to 0 for other types such as root ports, upstream/downstream ports |
| 8:6 | 0x0<br>RO | **Endpoint L0s Acceptable Latency (EPL0AL):** This field only applies to Endpoint and is hardwired to 0 for other types such as root ports, upstream/downstream ports |
| 5 | 0x1<br>RW/L | **Extended Tag Field Supported (ETFG):**<br>• 0: The PCIe supports 5-bit extended tag<br>• 1: The PCIe supports 8-bit extended tag<br><br>**Power Well:** PRST |
| 4:3 | 0x0<br>RO | **Phantom Functions Supported (PFS):** This field indicates the number of most significant bits of the function number portion of Requester ID in a TLP that are logically combined with the Tag identifier.<br>The PCIe cluster does not support phantom functions |
| 2:0 | 0x1<br>RW/L | **Max Payload size Supported (MPSS):** The PCIe SIP can tolerate up to 512-byte packets as maximum payload for project specific.<br>Defined encodings are:<br>• 000b 128 bytes max payload size<br>• 001b 256 bytes max payload size<br>• 010b 512 bytes max payload size<br>• 011b 1024 bytes max payload size<br>• 100b 2048 bytes max payload size<br>• 101b 4096 bytes max payload size<br>• 110b Reserved<br>• 111b Reserved<br><br>**Power Well:** PRST |

## 48.2.30 DEVCTL—Offset 48h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 48h

**Default:** 2000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x2<br>RW | **Max_Read_Request_Size (MRRS):** This field sets the maximum Read Requests size of the function as a requester.<br>The Function must not generate read requests with size exceeding the set value.<br>• 000b: 128 bytes maximum Read Request size<br>• 001b: 256 bytes maximum Read Request size<br>• 010b: 512 bytes maximum Read Request size<br>• 011b: 1024 bytes maximum Read Request size<br>• 100b: 2048 bytes maximum Read Request size<br>• 101b: 4096 bytes maximum Read Request size<br>• Others: Reserved Functions that do not generate Read Requests larger than 128B and functions that do not generate Read Requests on their own behalf are permitted to implement this field as Read Only (RO) with a value of 000b. |
| 11 | 0x0<br>RO | **Enable No Snoop (ENOSNP):** If this bit this is set, the function is permitted to set the No Snoop bit in the Requester attributes of transactions it initiates that do not require hardware enforced cache coherency. |
| 10 | 0x0<br>RO | **RSVD:** Reserved |
| 9 | 0x0<br>RO | **RSVD:** Reserved |
| 8 | 0x0<br>RW | **Extended Tag Field Enable (ETFE):** When set, this bit enables a function to use an 8-bit tag field as a Requester. Functions that do not implement this capability hardwire this bit to 0b. |
| 7:5 | 0x0<br>RW | **Maximum Payload Size (MPS):** This field sets maximum TLP payload size for the function.<br>As a receiver, the function must handle TLPs as large as the set value.<br>As a Transmitter, the function must not generate TLPs exceeding the set value.<br>• 000b: 128 bytes maximum payload size (default)<br>• 001b: 256 bytes maximum payload size<br>• 010b: 512 bytes maximum payload size<br>• 011b: 1024 bytes maximum payload size<br>• 100b: 2048 bytes maximum payload size<br>• 101b: 4096 bytes maximum payload size<br>• Others: Reserved |
| 4 | 0x0<br>RO | **Enable Relaxed Ordering (ENRO):** When set, the function is permitted to set the relaxed ordering bit in the attribute field of transactions it initiates that do not require strong write ordering. |
| 3 | 0x0<br>RW | **Unsupported Request Reporting Enable (URRE):** This bit controls the enabling of ERR_CORR, ERR_NONFATAL or ERR_FATAL messages on PCI Express for reporting 'Unsupported Request' errors. |
| 2 | 0x0<br>RW | **Fatal Error Reporting Enable (FERE):** When this bit is set, generation of the ERR_FATAL message is enabled. |
| 1 | 0x0<br>RW | **NonFatal Error Reporting Enable (NFERE):** When this bit is set, generation of the ERR_NONFATAL message is enabled. |
| 0 | 0x0<br>RW | **Correctable Error Reporting Enable (CERE):** When this bit is set, generation of the ERR_CORR message is enabled. |

## 48.2.31 DEVSTS—Offset 4Ah

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 4Ah

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO/V | **Transactions Pending (TP):** When set, this bit indicates that the function has issued Non-Posted Requests that have not been completed.<br>For Root or Switch port, it applies to Non-Posted Requests the port has issued on its own behalf (Port's Request ID). |
| 4 | 0x0<br>RO | **RSVD:** Reserved |
| 3 | 0x0<br>RW/1C/V | **Unsupported Request Detected (URD):** This bit indicates that this function received an unsupported request from PCI Express link.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 2 | 0x0<br>RW/1C/V | **Fatal Error Detected (FED):** This bit indicates that this function has detected a Fatal error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 1 | 0x0<br>RW/1C/V | **Non-Fatal Error Detected (NFED):** This bit indicates that this function has detected a Non-Fatal error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 0 | 0x0<br>RW/1C/V | **Correctable Error Detected (CED):** This bit indicates that this function has detected a Correctable error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |

## 48.2.32 LINKCAP—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 4Ch

**Default:** 09794883h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x9<br>RW/V/L | **Port Number (PN):** This field indicates the PCI Express port number assigned to this link.<br><br>**Power Well:** PRST |
| 23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x1<br>RW/L | **ASPM Optionality Compliance (ASPMOPCMP):** The ASPM Optionality Compliance bit was created as a tool to set clear expectations for hardware and software interaction.<br>This bit is Set to indicate hardware that conforms to the current specification.<br><br>**Power Well:** PRST |
| 21 | 0x1<br>RW/L | **Link Bandwidth Notification Capability (LBNC):** A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms.<br><br>**Power Well:** PRST |
| 20 | 0x1<br>RW/L | **Data Link Layer Active Error Reporting Capable (DLLERC):** For a Downstream Port, this bit must be hardwired to 1b if the component supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine.<br><br>**Power Well:** PRST |
| 19 | 0x1<br>RW/L | **Surprise Down Error Reporting Capable (SLDERC):** For a Downstream Port, this bit must be Set if the component supports the optional capability of detecting and reporting a Surprise Down error condition.<br><br>**Power Well:** PRST |
| 18 | 0x0<br>RO | **Clock Power Management Capable (CPMC):** A value of 0b indicates the component does not have this capability and that reference clock(s) must not be removed in these Link states. |
| 17:15 | 0x2<br>RW/L | **L1 Exit Latency (L1EL):** This field indicates the L1 exit latency for the given PCI-Express Link.<br>It indicates the length of time this port requires to complete transition from L1 to L0.<br>• 000: Less than 1us<br>• 001: 1us to less than 2us<br>• 010: 2us to less than 4us<br>• 011: 4us to less than 8us<br>• 100: 8us to less than 16us<br>• 101: 16us to less than 32us<br>• 110: 32us to less than 64us<br>• 111: More than 64us<br>If L1 is not supported, the value is undefined.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14:12 | 0x4<br>RW/L | **L0s Exit Latency (L0sEL):** This field indicates the L0s exit latency for the given PCI-Express Link. It indicates the length of time this port requires to complete transition from L0s to L0.<br>• 000b: Less than 64ns<br>• 001b: 64ns to less than 128ns<br>• 010b: 128ns to less than 256ns<br>• 011b: 256ns to less than 512ns<br>• 100b 512 ns to less than 1 us<br>• 101b: 1us to less than 2us<br>• 110b: 2us to less than 4us<br>• 111b: More than 4us<br>If L0s is not supported, the value is undefined.<br><br>**Power Well:** PRST |
| 11:10 | 0x2<br>RW/L | **ASPM Support (ASPMSUP):** This field indicates the level of ASPM supported on the given PCI Express Link.<br>• 00b: No ASPM Support<br>• 01b: L0s Supported<br>• 10b: L1 Supported<br>• 11b: L0s and L1 Supported<br><br>**Power Well:** PRST |
| 9:4 | 0x8<br>RO/V | **Maximum Link Width (MLW):** This field indicates the maximum link width implemented by the given PCI Express Link.<br>• 00h: Reserved<br>• 01h: x1<br>• 02h: x2<br>• 04h: x4<br>• 08h: x8<br>• 10h: x16<br>• 20h: x32 (Unsupported)<br>• Others: Reserved |
| 3:0 | 0x3<br>RO/V | **Maximum Link Speed (MLS):** This field indicates the supported link speed(s) of the associated port.<br>• 0001b: 2.5 Gb/s link speed is supported<br>• 0010b: 5.0 Gb/s and 2.5 Gb/s link speed supported<br>• 0011b: 8.0 Gb/s and 5.0 and 2.5 Gb/s link speed supported<br>• Others: Reserved. |

## 48.2.33 LINKCTL—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 50h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11 | 0x0<br>RW | **Link Autonomous Bandwidth Interrupt Enable (LABIE):** When Set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set.<br>Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. |
| 10 | 0x0<br>RW | **Link Bandwidth Management Interrupt Enable (LBMIE):** When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set.<br>This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. |
| 9 | 0x0<br>RW | **Hardware Autonomous Width Disable (HAWD):** Set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing Link width.<br>For a Multi-Function device associated with an Upstream Port, the bit in Function 0 is of type RW, and only Function 0 controls the components Link behavior.<br>In all other Functions of that device, this bit is of type RsvdP. Components that do not implement the ability autonomously to change Link width are permitted to hardwire this bit to 0b.<br>Default value of this bit is 0b. |
| 8 | 0x0<br>RW/L | **Enable Clock Power Management (ECPM):** Applicable only for Upstream Ports and with form factors that support a 'Clock Request' (CLKREQ#) mechanism, this bit operates as follows:<br>• 0b Clock power management is disabled and device must hold CLKREQ# signal low.<br>• 1b When this bit is Set, the device is permitted to use CLKREQ# signal to power manage Link clock according to protocol defined in appropriate form factor specification.<br>For a non-ARI multi-Function device, power-management-configuration software must only Set this bit if all Functions of the multi-Function device indicate a 1b in the Clock Power Management bit of the Link Capabilities register.<br>The component is permitted to use the CLKREQ# signal to power manage Link clock only if this bit is Set for all Functions.<br>For ARI Devices, Clock Power Management is enabled solely by the setting in Function 0.<br>The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the component.<br>The CLKREQ# signal may also be controlled via the L1 PM Substates mechanism.<br>Such control is not affected by the setting of this bit. Downstream Ports and components that do not support Clock Power Management (as indicated by a 0b value in the Clock Power Management bit of the Link Capabilities register) must hardwire this bit to 0b. Default value of this bit is 0b, unless specified otherwise by the form factor specification.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RW | **Extended Synch (ES):** When set, this bit forces extended transmission of 4096 FTS ordered sets in FTS and an extra 1024 TS1 at exit from L1 prior to entering L0.<br>This mode provides external devices monitoring the link time to achieve bit and symbol lock before the link enters L0 state and resumes communication. Default value for this bit is 0. |
| 6 | 0x0<br>RW | **Common Clock Configuration (CCCFG):** When set, this bit indicates that this component and the component at the opposite end of this link are operating with distributed common reference clocks.<br>A value of 0b indicates that this component and the component at the opposite end of this link are operating with asynchronous reference clock.<br>After changing the value in this bit in both components on a link, software must trigger the link to retrain by writing a 1b to the Retrain Link bit of the Downstream Port. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/V | **Retrain Link (RL):** When set, this bit initiates link retraining by directing the physical layer LTSSM to recovery state.<br>If the LTSSM is already in recovery or configuration, re-entering Recovery is permitted but not required.<br>Reads of this bit always return 0b.<br>This is used by the downstream ports, Root ports or DMI EP in RC only.<br>For the upstream port or virtual switch port, it is Read-only. |
| 4 | 0x0<br>RW | **Link Disable (LD):** This bit disables the Link by directing the LTSSM to the Disabled state when Set,<br>This is used by the downstream ports, Root ports or DMI EP in RC only.<br>For the upstream port or virtual switch port, it is Read-only. |
| 3 | 0x0<br>RO | **Read Completion Boundary (RCB):** This bit indicates the RCB value for Root Port, Endpoints and Bridges.<br>• 0b: 64 byte<br>• 1b: 128 byte<br>Configuration software must only set this bit if the Root Port Upstream from the Endpoint or Bridge reports an RCB value of 128 bytes (Root Port only) |
| 2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1:0 | 0x0<br>RW | **ASPM Control (ASPMCTL):** This field controls the level of ASPM enabled on a given PCI Express Link.<br>• 00b: Disabled<br>• 01b: L0s Entry Enabled<br>• 10b: L1 Entry Enabled<br>• 11b: L0s and L1 Entry Enabled<br>''L0s Entry Enabled' enables the Transmitter to enter L0s.<br>If L0s is supported, the Receiver must be capable of entering L0s even when the Transmitter is disabled from entering L0s (00b or 10b). |

## 48.2.34 LINKSTS—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 52h

**Default:** 1001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Link Autonomous Bandwidth Status (LABS):** This bit is set by hardware to indicate that the PCIe cluster has autonomously changed link or width, without the port transitioning through DL_Down status, for reason other than to attempt to correct unreliable link operation.<br>This bit must be set if the Physical Layer reports an autonomous speed or width change initiated by the Upstream Port. |
| 14 | 0x0<br>RW/1C/V | **Link Bandwidth Management Status (LBMS):** This bit is set by hardware to indicate that either of the following has occurred without the port transitioning through DL_Down status: |
| 13 | 0x0<br>RO/V | **Data Link Layer Link Active (DLLLA):** This bit indicates the status of the Data Link Control and Management Status Machine. |
| 12 | 0x1<br>RW/L | **Slot Clock Configuration (SCC):** When to 1b, this bit indicates that the component uses the same physical reference clock that he platform provides on the connector.<br>If the device uses an independent clock irrespective of the presence of a reference on the connector, this bit must be clear.<br>• 1: Indicates same reference clock.<br>• 0: Indicates independent reference clock<br><br>**Power Well:** PRST |
| 11 | 0x0<br>RO/V | **Link Training (LT):** This bit indicates that the Physical Layer LTSSM is in the Configuration or Recovery state |
| 10 | 0x0<br>RO | **Undefined:** Not applicable |
| 9:4 | 0x0<br>RO/V | **Negotiated Link Width (NLW):** This field indicates the negotiated width of the PCI Express link.<br>• 00 0001b: x1<br>• 00 0010b: x2<br>• 00 0100b: x4<br>• 00 1000b: X8<br>• 00 1100b: x12 not supported<br>• 01 0000b: x16<br>• 10 0000b: x32 not supported<br>• All other values are reserved.<br>The value in this field is undefined when the link is not up. |
| 3:0 | 0x1<br>RO/V | **Current Link Speed (CLS):** This field indicates the negotiated link speed of the given PCI Express link.<br>• 0001b: 2.5 Gb/s PCI Express Link<br>• 0010b: 5.0 Gb/s PCI Express Link<br>• 0011b: 8.0 Gb/s PCI Express Link<br>• Others: Reserved |

## 48.2.35 SLOTCAP—Offset 54h

Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 54h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RW/L | **Physical Slot Number (PSN):** This field indicates the physical slot number connected to the PCI-Express port.<br>It should be initialized to 0 for ports connected to devices that are either integrated on the system board or integrated within the same silicon such as the switch device or the Root port.<br><br>**Power Well:** PRST |
| 18 | 0x0<br>RW/L | **No Command Completed Support (NCCS):** When Set, this bit indicates that this slot does not generate software notification when an issued command is completed by the Hot-Plug Controller.<br>This bit is only permitted to be Set if the hot-plug capable Port is able to accept writes to all fields of the Slot Control register without delay between successive writes.<br><br>**Power Well:** PRST |
| 17 | 0x0<br>RW/L | **Electromechanical Interlock Present (EMIP):** This bit when set indicates that an Electromechanical Interlock is implemented on the chassis for this slot and that lock is controlled by bit 11 in Slot Control Register.<br><br>**Power Well:** PRST |
| 16:15 | 0x0<br>RW/L | **Slot Power Limit Scale (SPLS):** This field specifies the scale used for the Slot Power Limit Value.<br>Range of Values:<br>• 00: 1.0x<br>• 01: 0.1x<br>• 10: 0.01x<br>• 11: 0.001x<br>This register must be implemented if the Slot Implemented bit is set in EXPCAP. Writes to this register (SLOTCAP) when the Data Link Layer reports DL_Up status, or the link transitions from a non-DL_Up to a DL_Up would cause the Port to send the Set_Slot_Power_Limit Message.<br><br>**Power Well:** PRST |
| 14:7 | 0x0<br>RW/L | **Slot Power Limit Value (SPLV):** This field specifies the upper limit on power supplied by slot in conjunction with the Slot Power Limit Scale value defined previously Power limit (in Watts). i.e Power limit = SPLS x SPLV.<br>This register must be implemented if the Slot Implemented bit is set in EXPCAP.<br>Writes to this register (SLOTCAP) when the Data Link Layer reports DL_Up status, or the link transitions from a non-DL_Up to a DL_Up would cause the Port to send the Set_Slot_Power_Limit Message.<br><br>**Power Well:** PRST |
| 6 | 0x0<br>RW/L | **Hot-plug Capable (HPC):** This field defines hot-plug support capabilities for the PCI-Express port.<br>• 0: indicates that this slot is not capable of supporting Hot-plug operations.<br>• 1: indicates that this slot is capable of supporting Hot-plug operations<br><br>**Power Well:** PRST |
| 5 | 0x0<br>RW/L | **Hot-plug Surprise (HPS):** This field indicates that a device in this slot may be removed from the system without prior notification.<br>• 0: indicates that hot-plug surprise is not supported<br>• 1: indicates that hot-plug surprise is supported<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW/L | **Power Indicator Present (PIP):** This bit indicates that a Power Indicator is implemented on the chassis for this slot.<br>• 0: indicates that Power Indicator is not present<br>• 1: indicates that Power Indicator is present<br><br>**Power Well:** PRST |
| 3 | 0x0<br>RW/L | **Attention Indicator Present (AIP):** This bit indicates that an Attention Indicator is implemented on the chassis for this slot.<br>• 0: indicates that an Attention Indicator is not present<br>• 1: indicates that an Attention Indicator is present<br><br>**Power Well:** PRST |
| 2 | 0x0<br>RW/L | **MRL Sensor Present (MRLSP):** This bit indicates that an MRL Sensor is implemented on the chassis for this slot.<br>• 0: indicates that an MRL Sensor is not present<br>• 1: indicates that an MRL Sensor is present<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **Power Controller Present (PCP):** This bit indicates that a Power Controller is implemented on the chassis for this slot.<br>• 0: indicates that a Power Controller is not present<br>• 1: indicates that a Power Controller is present<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **Attention Button Present (ABP):** This bit indicates that an Attention Button is implemented on the chassis for this slot.<br>• 0: indicates that an Attention Button is not present<br>• 1: indicates that an Attention Button is present<br><br>**Power Well:** PRST |

## 48.2.36 SLOTCTL—Offset 58h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 58h

**Default:** 03C0h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13 | 0x0<br>RW | **Auto Slot Power Limit Disable (ASPLD):** When Set, this disables the automatic sending of a Set_Slot_Power_Limit Message when a Link transitions from a non-DL_Up status to a DL_Up status. |
| 12 | 0x0<br>RW | **Data Link Layer State Changed Enable (DLLSCE):** When set to 1, this field enables software notification when Data Link Layer Link Active reporting field is changed. |
| 11 | 0x0<br>RW/V | **Electromechanical Interlock Control (EIC):** When software writes either a 1 to this bit, The PCIe port pulses the EMIL pin per PCIExpress Server/Workstation Module Electromechanical Spec Rev 1.0. Write of 0 has no effect.<br>This bit always returns a 0 when read. If electromechanical lock is not implemented, then either a write of 1 or 0 to this register has no effect. |
| 10 | 0x0<br>RW | **Power Controller Control (PCC):** This bit indicates the current state of the Power applied to the slot of the PCIExpress port.<br>Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.<br>• 0: Power On<br>• 1: Power Off |
| 9:8 | 0x3<br>RW | **Power Indicator Control (PIC):** If a Power Indicator is implemented, writes to this register set the Power Indicator to the written state.<br>Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.<br>• 00: Reserved.<br>• 01: On<br>• 10: Blink (The PCIe cluster drives 1.5 Hz square wave for Chassis mounted LEDs in the case of legacy card form factor for PCI-Express devices)<br>• 11: Off When this register is written, the event is signaled via the virtual pins of the switch over a dedicated SMBus port.<br>The switch does not generated the Power_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software. |
| 7:6 | 0x3<br>RW | **Attention Indicator Control (AIC):** If an Attention Indicator is implemented, writes to this register sets the Attention Indicator to the written state.<br>Reads of this field reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined.<br>• 00: Reserved.<br>• 01: On<br>• 10: Blink (The switch drives 1.5 Hz square wave)<br>• 11: Off When this register is written, the event is signaled via the virtual pins of the switch over a dedicated SMBus port. |
| 5 | 0x0<br>RW | **Hot-plug Interrupt Enable (HPIE):** This field enables the generation of Hot-plug interrupts and events in the PCIExpress port.<br>• 0: Disables Hot-plug events and interrupts<br>• 1: Enables Hot-plug events and interrupts |
| 4 | 0x0<br>RW | **Command Completed Interrupt Enable (CCIE):** This field enables the generation of Hot-plug interrupts when a command is completed by the Hot-plug controller connected to the PCI-Express port.<br>• 0: Disables hot-plug interrupts on a command completion by a hot-plug Controller<br>• 1: Enables hot-plug interrupts on a command completion by a hot-plug Controller |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW | **Presence Detect Changed Enable (PDCIE):** This bit enables the generation of hot-plug interrupts or wake messages via a presence detect changed event.<br>• 0: Disables generation of hot-plug interrupts or wake messages when a presence detect changed event happens.<br>• 1- Enables generation of hot-plug interrupts or wake messages when a presence detect changed event happens. |
| 2 | 0x0<br>RW | **MRL Sensor Changed Enable (MSCIE):** This bit enables the generation of hot-plug interrupts or wake messages via a MRL Sensor changed event.<br>• 0: Disables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens.<br>• 1: Enables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens. |
| 1 | 0x0<br>RW | **Power Fault Detected Enable (PFDIE):** This bit enables the generation of hot-plug interrupts or wake messages via a power fault event.<br>• 0: Disables generation of hot-plug interrupts or wake messages when a power fault event happens.<br>• 1: Enables generation of hot-plug interrupts or wake messages when a power fault event happens. |
| 0 | 0x0<br>RW | **Attention Button Pressed Enable (ABPE):** This bit enables the generation of hot-plug interrupts or wake messages via an attention button pressed event.<br>• 0: Disables generation of hot-plug interrupts or wake messages when the attention button is pressed.<br>• 1: Enables generation of hot-plug interrupts or wake messages when the attention button is pressed |

## 48.2.37 SLOTSTS—Offset 5Ah

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 5Ah

**Default:** 0040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RW/1C/V | **Data Link Layer State Changed (DLLSCS):** This bit is set (if it is not already set) when the state of the Data Link Layer Link Active bit in the Link Status register has changed.<br>Software must read Data Link Layer Active field to determine the link state before initiating configuration cycles to the hot plugged device. |
| 7 | 0x0<br>RO/V | **Electromechanical Interlock Status (ELS):** A read to this register returns the current state of the Electromechanical Interlock (the EMILS pin) which has the defined encodings as:<br>• 0: Electromechanical Interlock Disengaged<br>• 1: Electromechanical Interlock Engaged |
| 6 | 0x1<br>RO/V | **Presence Detect State (PDS):** This field conveys the Presence Detect status of an adapter in the slot.<br>It reflects the Logical 'OR' of the Physical Layer in-band presence detect mechanism and any out-of-band presence detect mechanism defined for the slot's corresponding form factor.<br>• 0: Card/Module/Cable slot empty or Cable Slot occupied but not powered<br>• 1: Card/module Present in slot (powered or unpowered) or cable present and powered on the other end.<br>This field is set to 1 by the switch following reset for all the ports since it is related to the complement of the EXPCAP.SI bit whose default is 0 implying that the card is connected to the switch on the motherboard.<br>Software should ensure that for those standard PCI-Express ports which are hot-plug capable, it sets the EXPCAP.SI field to '1' such that PDS becomes 0.<br>Then any subsequent hotplug action will ensure that this field is set. Refer to the algorithm given below:<br>if (EXPCAP.SI == FALSE) // always set to 1b and ignore the input from inband/OOB presence detect SLOTSTS.PDS = 1b else //<br>if (EXPCAP.SI == TRUE) // Slot Empty = 0, Card Present = 1 SLOTSTS.PDS reflects slot status of side-band SMBUS Presence Detect signal OR Inband Presence Detect Mechanism. |
| 5 | 0x0<br>RO/V | **MRL Sensor State (MRLSS):** This bit reports the status of an MRL sensor if it is implemented.<br>• 0: MRL Closed<br>• 1: MRL Open<br>If Hotplug is implemented through VPP, the correct polarity of the external MRL sensor in the motherboard should be translated to map to the encodings as defined in this register field.<br>(Tylersburg issue) |
| 4 | 0x0<br>RW/1C/V | **Command Completed (CCS):** This bit is set when the hot-plug controller completes an issued command and is ready to accept a new command.<br>It is subsequently cleared by software after the field has been read and processed.<br>If Command Completed notification is supported, then the No Command Completed Support bit in the Slot Capabilities register should be 0. |
| 3 | 0x0<br>RW/1C/V | **Presence Detect Changed (PDCS):** This bit is set when a Presence Detect Changed event is detected.<br>It is subsequently cleared by software after the field has been read and processed.<br>On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect.<br>RPs may hold the re-training to occur until the SLOTSTS.PDCS bit has been cleared by software for the removal event. If the device is configured as a slot in the EXPCAP_.SI bit, this bit will need to be cleared before re-training will occur. |
| 2 | 0x0<br>RW/1C/V | **MRL Sensor Changed (MRLSC):** This bit is set when an MRL Sensor Changed event is detected.<br>It is subsequently cleared by software after the field has been read and processed.<br>On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/1C/V | **Power Fault Detected (PFD):** This bit is set when a power fault event is detected by the power controller.<br>It is subsequently cleared by software after the field has been read and processed.<br>On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect. |
| 0 | 0x0<br>RW/1C/V | **Attention Button Pressed (ABP):** This bit is set when the attention button is pressed.<br>It is subsequently cleared by software after the field has been read and processed.<br>On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect.<br>Design should implement this as an edge detection logic in the chipset/platform to recognize a 0 -) 1 transition as a ABP event.<br>Due to the long time frames for human interaction, software could potentially have serviced and cleared this event while the operator is pressing the attention button.<br>For instance, if the user presses the Attention button that lasts for 2ms and the VPP sampling rate is, say 100us and if the Interrupt Service routine (ISR) processes and clears the ABP in 100us, there could be up to 10 (i.e 2000/(100+100)) different interrupts for the same event.Hardware must only generate 1 interrupt for every ABP event by treating it as edge based rather than level based event. |

## 48.2.38 ROOTCTL—Offset 5Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 5Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:5 | 0x0<br>RO | **RSVD0:** Reserved |
| 4 | 0x0<br>RW | **CRS Software Visibility Enable (CRSSVE):** This bit, when set, enables the Root Port to return Configuration Retry Status (CRS) Completion status to software.<br>• 1: Enable software to received a CRS status. This allows software to make the decision to re-issue the configuration request or move on and re-issue the request at a later time.<br>• 0: Disable software from receiving a CRS response. The PCIe will wait and re-issue the configuration request until it receives a response other than CRS without notifying software. CRS is implemented for general Root ports. (See 'ROOTCAP'). |
| 3 | 0x0<br>RW | **PME Interrupt Enable (PMEIE):** This field controls the generation of interrupts for PME messages.<br>• 1: Enables interrupt generation upon receipt of a PME message as reflected in the PME Status bit defined in the ROOTSTS register. A PME interrupt is generated if the PMESTATUS register bit defined i 'Root Status Register "ROOTSTS—Offset 60h" on page 1466, is set when this bit is set from a cleared state.<br>• 0: Disables interrupt generation for PME messages. |
| 2 | 0x0<br>RW | **System Error on Fatal Error Enable (SEFEE):** This field controls generation of system errors in the PCI-Express port hierarchy for fatal errors.<br>• 1: indicates that a System Error should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port.<br>• 0: No System Error should be generated on a fatal error (ERR_FATAL) reported by any of the devices in the hierarchy. |
| 1 | 0x0<br>RW | **System Error on Non-Fatal Error Enable (SENFEE):** This field controls generation of system errors in the PCI-Express port hierarchy for non-fatal errors.<br>• 1: indicates that a System Error should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port.<br>• 0: No System Error should be generated on a non-fatal error (ERR_NONFATAL) reported by any of the devices in the hierarchy. |
| 0 | 0x0<br>RW | **System Error on Correctable Error Enable (SECEE):** This field controls generation of system errors in the PCI-Express port hierarchy for correctable errors.<br>• 1: indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with and including this PCI-Express port.<br>• 0: No System Error should be generated on a correctable error (ERR_COR) reported by any of the devices in the hierarchy associated with and including this PCI-Express port. |

## 48.2.39 ROOTCAP—Offset 5Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 5Eh

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/L | **CRS Software Visibility (CRSSV):** This bit, when set, indicates that the Root Port is capable of returning Configuration Retry Status (CRS) on completions to software.<br>Based on Design input, the CRS notification will be implemented in the EIP PCIe macro and hence the default is set to 1.<br>The field attribute is made as RW-O for any specific changes to be installed by BIOS<br><br>**Power Well:** PRST |

## 48.2.40 ROOTSTS—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RO/V | **PME Pending (PMEPEND):** This field indicates that another PME is pending when the PME Status bit is set.<br>When the PME Status bit is cleared by software, the pending PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately.<br>The PME pending bit is cleared by hardware if no more PMEs are pending.<br>***Note:*** The root port can handle two outstanding PM_PME messages in its internal queues of the Power Management controller per port. If the downstream device issues more than 2 PM_PME messages successively, it will be dropped. |
| 16 | 0x0<br>RW/1C/V | **PME Status (PMESTS):** This field indicates status of a PME that is underway in the PCI-Express port.<br>1: PME was asserted by a requester as indicated by the PMEREQID field.<br>This bit is cleared by software by writing a '1'. Subsequent PMEs are kept pending until the PME Status is cleared. |
| 15:0 | 0x0<br>RO/V | **PME Requester ID (PMERID):** This field indicates the PCI requester ID of the last PME requestor. |

## 48.2.41 DEVCAP2—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 64h

**Default:** 00000437h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:22 | 0x0<br>RW/L | **Max End-End TLP Prefixes (MAXE2ETLP):** Indicates the maximum number of End-End TLP Prefixes supported by this Function.<br>TLPs received by this Function that contain more End-End TLP Prefixes than are supported must be handled as Malformed TLPs.<br>• 01: 1 End-End TLP Prefix<br>• 10: 2 End-End TLP Prefixes<br>• 11: 3 End-End TLP Prefixes<br>• 00: 4 End-End TLP Prefixes<br>• If End-End TLP Prefix Supported is Clear, this field is Reserved.<br><br>**Power Well:** PRST |
| 21 | 0x0<br>RW/L | **End-End TLP Prefix Supported (E2ETLPS):** Indicates whether End-End TLP Prefix support is offered by a Function.<br>• 0: No Support<br>• 1: Support is provided to receive TLPs containing End-End TLP Prefixes.<br>The EIP does not support E2E TLP prefix and is hardwired to 0.<br><br>**Power Well:** PRST |
| 20 | 0x0<br>RW/L | **Extended Fmt Field Supported (EXFMTFS):**<br>• 0: The Function supports a 2-bit definition of the Fmt field (default)<br>• 1: The Function supports the 3-bit definition of the Fmt field.<br>Must be Set for Functions that support End-End TLP Prefixes.<br><br>**Power Well:** PRST |
| 19:18 | 0x0<br>RW/L | **OBFF Supported (OBFFS):**<br>• 00b: OBFF Not Supported<br>• 01b: OBFF supported using Message signaling only<br>• 10b: OBFF supported using WAKE# signaling only<br>• 11b: OBFF supported using WAKE# and Message signaling<br>Applicable only to Root Ports, Switch Ports, and Endpoints that support this capability. Must be 00b for other function types.<br><br>**Power Well:** PRST |
| 17:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13:12 | 0x0<br>RW/L | **TPH Completer Supported (TPHCS):** Value indicates Completer support for TPH or Extended TPH.<br>• 00b:TPH and Extended TPH Completer not supported<br>• 01b: TPH Completer supported, Extended TPH completer not supported<br>• 10b: Reserved<br>• 11b: Both TPH and Extended TPH Completer supported<br>Applicable only to Root Ports and Endpoints. Must be 00b for other function types.<br><br>**Power Well:** PRST |
| 11 | 0x0<br>RW/L | **LTR Mechanism Supported (LTRMS):** This bit when set indicates support for the Latency Tolerance Reporting (LTR) mechanism.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW/L | **No RO-enabled PR-PR Passing (NROEPRPASS):** This bit if set to 1b disables passing permitted by row entry A2b in PCIe ordering table that is associated with the Relaxed Ordering Attribute field being set.<br>This bit applies only for Switches and RCs that support peer-to-peer traffic between ports.<br>The bit only applies to posted requests being forwarded through the switch or RC.<br>It does not apply to traffic originating or terminating with in the switch or RC itself.<br><br>**Power Well:** PRST |
| 9 | 0x0<br>RW/L | **AD128 CAS Completer Supported (AD128ACS):** This bit must be set to 1b if the function supports this optional capability. Applicable to functions with memory space BARs as well as Root Ports.<br><br>**Power Well:** PRST |
| 8 | 0x0<br>RW/L | **AD64-bit AtomicOp Completer Supported (AD64ACS):** This bit must be set to 1b if the function supports this optional capability.<br>Applicable to Functions with Memory Space BARs as well as all Root Ports, must be 0b otherwise.<br>Includes FetchAdd, Swap, and CAS AtomicOps.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RW/L | **AD32 bit AtomicOp Completer Supported (AD32ACS):** Applicable to Functions with Memory Space BARs as well as all Root Ports, must be 0b otherwise.<br>Includes FetchAdd, Swap, and CAS AtomicOps.<br>This bit must be set to 1b if the Function supports this optional capability. This is set to 1 for Rootport implementations in the PCIe EIP.<br><br>**Power Well:** PRST |
| 6 | 0x0<br>RW/L | **AtomicOp Routing Supported (ARS):** This bit must be set to 1b if the port supports this optional capability.<br>Applicable only to switch upstream ports, Switch downstream ports, and root ports.<br>Must be 00b for other function types.<br><br>**Power Well:** PRST |
| 5 | 0x1<br>RW/L | **ARI Forwarding Supported (ARI):** This bit is set to 1b indicating that the downstream port supports this capability.<br>Applicable only to Switch Downstream Ports and Root Ports, must be 0b for other Function types.<br>This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability.<br><br>**Power Well:** PRST |
| 4 | 0x1<br>RW/L | **Completion Timeout Disable Support (CTDS):** A value of 1b indicates support for the completion Timeout Disable Mechanism.<br>Support of completion timeout disable is optional for Root Ports.<br>The PCIe port supports completions timeout disable.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x7<br>RW/L | **Completion Timeout Range Supported (CTRS):** This field indicates device support for the optional Completion Timeout programmability mechanism.<br>This mechanism allows system software to modify the Completion Timeout value.<br>This field is applicable only to Root Ports, Endpoints that issue requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of request issues on PCI Express.<br>Four time values ranges are defined:<br>• Range A: 50us to 10ms<br>• Range B: 10ms to 250ms<br>• Range C: 250ms to 4s<br>• Range D: 4s to 64s<br>Bits ares set according to table below to show timeout value ranges supported.<br>• 0000b: Completions Timeout programming not supported -- values are fixed by implementation in the range 50us to 50ms.<br>• 0001b: Range A<br>• 0010b: Range B<br>• 0011b: Range A and B<br>• 0110b: Range B and C<br>• 0111b: Range A, B, and C<br>• 1111b: Range A, B, C and D All other values are reserved.<br>• For all other devices this field is reserved and maybe be hardwired to 0000b.<br><br>**Power Well:** PRST |

## 48.2.42 DEVCTL2—Offset 68h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 68h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **End-End TLP Prefix Blocking (E2ETLPBLK):** Controls whether the routing function is permitted to forward TLPs containing an End-End TLP Prefix.<br>• 0: Forwarding Enabled - Function is permitted to send TLPs with End-End TLP Prefixes.<br>• 1: Forwarding Blocked - Function is not permitted to send TLPs with End-End TLP Prefixes. |
| 14:13 | 0x0<br>RW | **OBFF Enable (OBFFE):**<br>• 00b: OBFF Disabled<br>• 01b: OBFF Enabled using Message signaling [Variation A]<br>• 10b: OBFF Enabled using Message signaling [Variation B]<br>• 11b: OBFF Enabled using WAKE# signaling<br>This field is required for all Ports that support the OBFF Capability. For a Multi-Function Device associated with an Upstream Port of a Device that implements OBFF, the field in Function 0 is of type RW, and only Function 0 controls the Component's behavior. In all other Functions of that Device, this field is of type RsvdP. |
| 12:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RW/V | **LTR Mechanism Enable (LTRME):** When set to 1b, this bit enables Upstream Ports to send LTR messages and Downstream Ports to process LTR Messages.<br>Applicable to Root Ports, Switches, and Endpoints that implement the LTR capability must implement this bit.<br>For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status. |
| 9 | 0x0<br>RO | **IDO Completion Enable (IDOCE):** This bit when set to 1b allows the function to set the ID-Based Ordering (IDO) bit of completions it returns.<br>Applicable only to Endpoints including RC integrated Endpoints and Root Ports. |
| 8 | 0x0<br>RO | **IDO Request Enable (IDORE):** This bit when set to 1b allows the function to set the ID-Based Ordering (IDO) bit on requests it initiates.<br>Applicable only to Endpoints including RC integrated Endpoints and Root Ports. |
| 7 | 0x0<br>RW | **AtomicOp Egress Blocking (AEB):** When this bit is set, AtomicOp requests that target this out going Egress port must be blocked.<br>Applicable and mandatory for switch Upstream Ports, Switch Downstream Ports, and Root Ports that implement AtomicOp routing capability. Otherwise, must be hardwired to 0b. |
| 6 | 0x0<br>RW | **AtomicOp Requester Enable (ARE):** Applicable only to Endpoints and Root Ports, must be hardwired to 0b for other Function types.<br>The Function is allowed to initiate AtomicOp Requests only if this bit and the Bus Master Enable bit in the Command register are both Set. |
| 5 | 0x0<br>RW | **ARI Forwarding Enable (ARIE):** When set to 1b, ARI is enabled for the downstream port or root ports.<br>The Downstream Port disables its traditional Device Number field being 0 enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port. Default value of this bit is 0b. |
| 4 | 0x0<br>RW | **Completion Timeout Disable (CTD):**<br>• 1: Disable the completions timeout mechanism for all NP transactions.<br>• 0: Completion timeout is enabled for all NP transactions. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RW | **Completion Timeout Value (CTV):** In devices that support completion timeout programmability, this field allows system software to modify the completion timeout range.<br>The following encodings and corresponding timeout ranges are defined based on implementation choice.<br>This field provides finer control based over the ranges chosen by DEVAP2.CTRS.<br>The implementation specific range is shown in brackets<br>• 0000b: 50us to 50ms (16.8ms - 25.2ms based on core clk period)<br>• 0001b: 50us to 100us (65.5us - 99.3us based on core clk period)<br>• 0010b: 1ms to 10ms (4.2ms - 6.3ms based on core clk period)<br>• 0101b: 16ms to 55ms (33.6ms - 50.3ms based on core clk period)<br>• 0110b: 65ms to 210ms (134.2ms - 201.3ms based on core clk period)<br>• 1001b: 260ms to 900ms (536.9ms - 805.3ms based on core clk period)<br>• 1010b: 1s to 3.5s (2.1s - 3.2s based on core clk period)<br>• 1101b: 4s to 13s (8.6s - 12.9s based on core clk period)<br>• 1110b: 17s to 64s (34.4s - 51.5s based on core clk period)<br>• All others are reserved.<br>It is highly recommended that the completion timeout value not be less then 10ms.<br>A small completion timeout value may result in premature completion timeout for slower responding devices. |

## 48.2.43 LINKCAP2—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 6Ch

**Default:** 0000000Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22:16 | 0x0<br>RW/L | **Lower SKP OS Reception Supported Speeds Vector (LSKPOSRSSV):** If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports receiving SKP OS at the rate defined for SRNS while running in SRIS.<br>Bit definitions within this field are:<br>• Bit 0 2.5 GT/s<br>• Bit 1 5.0 GT/s<br>• Bit 2 8.0 GT/s<br>• Bits 6:3 RsvdP<br>Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions. Behavior is undefined if a bit is Set in this field and the corresponding bit is not Set in the Supported Link Speeds Vector.<br>Currently, design does not support reception of low frequency SOS in L0.<br><br>**Power Well:** PRST |
| 15:9 | 0x0<br>RW/L | **Lower SKP OS Generation Supported Speeds Vector (LSKPOSGSSV):** If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports software control of the SKP Ordered Set transmission scheduling rate.<br>Bit definitions within this field are:<br>• Bit 0 2.5 GT/s<br>• Bit 1 5.0 GT/s<br>• Bit 2 8.0 GT/s<br>• Bits 6:3 RsvdP<br>Multi-Function devices associated with an Upstream Port must report the same value in this field for all Functions. Behavior is undefined if a bit is Set in this field and the corresponding bit is not Set in the Supported Link Speeds Vector.<br><br>**Power Well:** PRST |
| 8 | 0x0<br>RW/L | **Crosslink Supported (CLS):**<br>When set to 1b, this bit indicates that the associated Port supports crosslinks<br>When set to 0b on a Port that supports Link speeds of 8.0 GT/s or higher, this bit indicates that the associated Port does not support crosslinks.<br>When set to 0b on a Port that only supports Link speeds of 2.5 GT/s or 5.0 GT/s, this bit provides no information regarding the Port's level of crosslink support.<br>It is recommended that this bit be Set in any Port that supports crosslinks even though doing so is only required for Ports that also support operating at 8.0 GT/s or higher Link speeds.<br>***Note:*** Software should use this bit when referencing fields whose definition depends on whether or not the Port supports crosslinks.<br><br>**Power Well:** PRST |
| 7:1 | 0x7<br>RO/V | **Supported Link Speeds Vector (SLNKSPV):** This field indicates the supported Link speed(s) of the associated Port.<br>For each bit, a value of 1b indicates that the corresponding Link speed is supported, otherwise, the Link speed is not supported.<br>Bit definitions are:<br>• Bit 1 2.5 GT/s<br>• Bit 2 5.0 GT/s<br>• Bit 3 8.0 GT/s<br>• Others: Reserved |
| 0 | 0x0<br>RO | **RSVD1:** Reserved |

## 48.2.44 LINKCTL2—Offset 70h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 70h

**Default:** 0003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RW/P | **Compliance Preset/De-emphasis (CD):** For 8 GT/s and 5 GT/s: This bit sets the transmitter preset level for Polling.<br>Compliance state if the entry occurred due to the Enter Compliance bit being 1b.<br>The Encoding are defined as follows:<br>8 GT/s Rate:<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>5 GT/s Rate:<br>• 001b: -3.5dB<br>• 000b: -6 dB<br>• All other are reserved.<br>When the link is operating at 2.5 Gb/s, the setting of this bit has no effect.<br>Components that support only 2.5 GT/s speed are permitted to hardwire this bit to 0b. |
| 11 | 0x0<br>RW/P | **Compliance SOS (CSOS):** When set to 1b, the LTSSM is required to send Skip Ordered Sets periodically in between the (modified) compliance patterns. |
| 10 | 0x0<br>RW/P | **Enter Modified Compliance (EMC):** When set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.<br>Compliance substate. This register is intended for debug, compliance testing purposes only.<br>System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value. |
| 9:7 | 0x0<br>RW/V/P | **Transmit Margin (TM):** This field controls the value of the non-de-emphasized voltage level at the transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.<br>Configuration substate.<br>• 000b: Normal operating range<br>• 001b: 800-1200mV for full swing and 400-700mV for half-swing<br>• 010b-110b: TBD<br>• Others: Reserved<br>This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing.<br>In all other cases, the system must ensure that this register is set to the default value. |
| 6 | 0x0<br>RW/L | **Selectable De-emphasis (SD):** Applicable to RootPorts and downstream ports of PCI Express to PCIe bridges.<br>When the link is operating at 5 Gb/s speed, this bit selects the level of de-emphasis for an Downstream Port.<br>Encodings: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/P | **Hardware Autonomous Speed Disable (HASD):** When set, this bit disables hardware from changing the link speed for device specific reasons other than attempting to correct unreliable link operation by reducing link speed for device-specific reasons other than attempting to correct unreliable link operations by reducing link speed.<br>Initial transition to the highest supported common link speed is not blocked by this bit. |
| 4 | 0x0<br>RW/V/P | **Enter Compliance (EC):** Software is permitted to force a link to enter compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link. |
| 3:0 | 0x3<br>RW/V/P | **Target Link Speed (TLS):** For downstream ports and root ports, this field sets an upper limit on link operational speed by restricting the values advertised by the Downstream Port in its training sequences.<br>• 0001b: 2.5 Gb/s Target Link Speed<br>• 0010b: 5.0 Gb/s Target Link Speed<br>• 0011b: 8.0 Gb/s Target Link Speed<br>• Others: Reserved. The encoding is the binary value of the bit in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the desired target Link speed.<br>• All other encodings are reserved<br>If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, the result is undefined.<br>The default value of this field is the highest Link speed supported by the component (as reported in the Max Link Speed field of the Link Capabilities register) unless the corresponding platform/form factor requires a different default value.<br>For both Upstream and Downstream Ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a Link into compliance mode. |

## 48.2.45 LINKSTS2—Offset 72h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 72h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RW/1C/V/P | **Link Equalization Request (LNKEQREQ):** This bit is set by hardware to request the Link equalization process to be performed on the Link. |
| 4 | 0x0<br>RO/V/P | **Equalization Phase 3 Successful (EQPH3SUCC):** When set to 1b, this bit indicates that Phase 3 of the Transmitter Equalization procedure has successfully completed. |
| 3 | 0x0<br>RO/V/P | **Equalization Phase 2 Successful (EQPH2SUCC):** When set to 1b, this bit indicates that Phase 2 of the Transmitter Equalization procedure has successfully completed. |
| 2 | 0x0<br>RO/V/P | **Equalization Phase 1 Successful (EQPH1SUCC):** When set to 1b, this bit indicates that Phase 1 of the Transmitter Equalization procedure has successfully completed. |
| 1 | 0x0<br>RO/V/P | **Equalization Complete (EQCMPLT):** When set to 1b, this bit indicates that the Transmitter Equalization procedure has successfully completed. |
| 0 | 0x0<br>RO/V | **Current De-emphasis Level (CDL):** When the link is operating at 5 Gb/s speed, this bit reflects the level of de-emphasis.<br>• 1b: -3.5 dB<br>• 0b: -6dB<br>The value in this bit is undefined when the Link is not operating at 5.0 GT/s speed.<br>Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 0b. |

## 48.2.46 PMCAPLST—Offset 80h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 80h

**Default:** 8801h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x88<br>RO | **Next Pointer (NextPointer):** Contains the offset of the next item in the capabilities list. (SSCAPLST) |
| 7:0 | 0x1<br>RO | **Capability ID (CapabilityID):** Identifies the function as PCI Power Management capable. |

## 48.2.47 PMCAP—Offset 82h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 82h

**Default:** C803h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x19<br>RO | **PME_Support (PMES):** PME assertion is supported when in D3hot/D3cold.<br>Identifies power states in which the PCIe can send an 'Assert_PMEGPE/Deassert PMEGPE' message.<br>Bits 15, 14 and 11 must be set to '1' for PCIPCI bridge structures representing ports on root complexes.<br>The definition of these bits is taken from the PCI Bus Power Management Interface Specification Revision 1.1.<br>• XXXX1b - Assert_PMEGPE/Deassert PMEGPE can be sent from D0<br>• XXX1Xb - Assert_PMEGPE/Deassert PMEGPE can be sent from D1 (Not supported by PCI-E)<br>• XX1XXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D2 (Not supported by PCI-E)<br>• X1XXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 hot (Supported by PCI-E)<br>• 1XXXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 cold (Supported by PCI-E)<br>In the EIP PCIe implementation, there is no PME support for D3 cold as the part does not have the capability. For PCIe compliance purposes, bit 15 is set to 1. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** Not supported |
| 9 | 0x0<br>RO | **D1 Support (D1S):** Not supported |
| 8:6 | 0x0<br>RO | **Auxiliary Current (AC):** Auxiliary power is not supported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Device-specific initialization is not required when transitioning to D0 from D3hot state. This bit is zero. |
| 4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x0<br>RO | **PME Clock (PMECLK):** Does not apply to PCI Express. Hard-wired to 0. |
| 2:0 | 0x3<br>RO | **Version (VER):** The PM implementation in the PCIe cluster is compliant with PCI Bus Power Management Interface Specification, Revision 1.2. |

## 48.2.48 PMCSR—Offset 84h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 84h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V/P | **PME Status (PMESTS):** Reserved. |
| 14:13 | 0x0<br>RO | **Data Scale (DC):** Not supported |
| 12:9 | 0x0<br>RO | **Data Select (DS):** Not supported |
| 8 | 0x0<br>RW/P | **PME Enable (PMEEN):** Gates assertion of the PME message. |
| 7:4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x1<br>RW/L | **No Soft Reset (NSR):** This bit when 1b indicates that a device transitioning from D3hot to D0 does not perform an internal reset. The configuration context is preserved.<br><br>**Power Well:** PRST |
| 2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1:0 | 0x0<br>RW/V | **Power State (PS):** This field is used both to determine the current power state of a function and to set the function into a new power state.<br>The definition of the supported values is given below:<br>• 0h - D0<br>• 3h - D3hot<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. |

## 48.2.49 PMBSE—Offset 86h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 86h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Bus Power/Clock Control Enable (BPCC_EN):** Neither bus or clock control of PCI is supported when in D3hot state. This bit is hard-wired to 0. |
| 6 | 0x0<br>RO | **B2/B3# (B23EN):** Not supported. This bit has no meaning since the BPCC_En bit is hard-wired to 0. |
| 5:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 48.2.50 SSCAPLST—Offset 88h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 88h

**Default:** 900Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x90<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list. (MSICAPLST) |
| 7:0 | 0xd<br>RO | **Capability ID (CAPID):** Identifies the function as Subsystem Identification capable. |

## 48.2.51 SSVID—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8Ch

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RW/L | **Subsystem Vendor Identifier (SSVID):** Assigned by PCI-SIG for vendor ID.<br><br>**Power Well:** PRST |

## 48.2.52 SSID—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **Subsystem Identifier (SSID):** Assigned to uniquely identify the subsystem vendor.<br><br>**Power Well:** PRST |

## 48.2.53 MSICAPLST—Offset 90h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 90h

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list (MSIXCAPLST). A null value is used to indicate that this is the last capability. The Next Pointer is terminated for DP, vRP/vSP, RP and DMI EP in RC. |
| 7:0 | 0x5<br>RO | **Capability ID (CAPID):** Identifies the function as MSI capable. |

## 48.2.54 MSICTL—Offset 92h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 92h

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x1<br>RW/L | **Per-vector masking capable (PVM):** This bit indicates that PCI Express ports support MSI per-vector masking.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **Address 64-Bit Capable (AD64C):** When set, this bit indicates that the function is capable of generating a 64-bit message address. |
| 6:4 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Only one message is supported.<br>These bits are R/W for software compatibility. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Only one message is supported. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. |

## 48.2.55 MSIADDR—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address:** Message address specified by the system, always DWORD aligned |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 48.2.56 MSIDATA—Offset 98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>• 0: Edge Triggered<br>• 1: Level Triggered |
| 14 | 0x0<br>RW | **Level (LVL):** if TM is 0h, then this field is a don't care.<br>Edge triggered messages are always treated as assert messages.<br>For level triggered interrupts, this bit reflects the state of the interrupt input if TM is 1h, then<br>• 0: Deassert Messages<br>• 1: Assert Messages |
| 13:12 | 0x0<br>RW | **Future:** These bits are don't care for an IOxAPIC interrupt message data field specification. |
| 11:8 | 0x0<br>RW | **Delivery Mode (DM):**<br>• 0000: Fixed: Trigger Mode can be edge or level.<br>• 0001: Lowest Priority: Trigger Mode can be edge or level.<br>• 0010: SMI/PMI/MCA - Not supported via MSI of root port<br>• 0011: Reserved - Not supported via MSI of root port<br>• 0100: NMI - Not supported via MSI of root port<br>• 0101: INIT - Not supported via MSI of root port<br>• 0110: Reserved<br>• 0111: ExtINT - Not supported via MSI of primary port<br>• 1000 -1111 - Reserved |
| 7:0 | 0x0<br>RW | **Interrupt Vector (IV):** The interrupt vector will be modified by the EIP PCIe Gen3 to provide context sensitive interrupt information for different events that require attention from the processor. e.g Hot plug, Power Management and RAS error events. |

## 48.2.57 MSIMSK—Offset 9Ch

The MSI Mask Bit register enables software to disable message sending on a per-vector basis.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RW | **Mask Bits (MSKB):** For each Mask bit that is set, the PCI Express port is prohibited from sending the associated message. Corresponding bits are masked if set to '1' |

## 48.2.58 MSIPENDING—Offset A0h

The Mask Pending register enables software to defer message sending on a per-vector basis.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x0<br>RO/V | **Pending Bits (PB):** For each Pending bit that is set, the PCI Express port has a pending associated message. Corresponding bits are pending if set to '1' |

## 48.2.59 RPPMCSR—Offset A8h

bek_eip_pcie3_regs_usx16.fm bek_eip_pcie3_regs_usx8.fm
bek_eip_pcie3_regs_vsp.fm cnx_eip_pcie3_regs_dmix4.fm
cnx_eip_pcie3_regs_rp4px4.fm cnx_eip_pcie3_regs_rp4px16.fm
cnx_eip_pcie3_regs_ep.fm cnx_eip_pcie3_regs_rlusx16.fm
cdf_eip_pcie3_regs_rldsx16.fm dnv_eip_pcie3_regs_rp4px8.fm
dnv_eip_pcie3_regs_rp4px16.fm dnv_eip_pcie3_regs_rp4px4.fm
dnv_eip_pcie3_regs_vrp.fm

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A8h

**Default:** 00000020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RO/V | **Mod-PHY Core Well Power Gating Status (MCWPGS):** This field reflects the effective 'Logical OR' state of the mod-PHY data lane 'Core Power Request' from controller to PHY and 'Core Power Ack' from PHY to controller for all the data lanes owned by the port.<br>Software can read this bit to know that the mod-PHY Core Well power gating handshake is completed, before it proceeds to power down the mod-PHY Suspend Well using the RPPGEN.MDLSWPR and RPPGEN.MDLSWPA bits.<br>This bit is intended to be used for RTD3, as part of the flow to power gate the mod-PHY Core Well. |
| 7 | 0x0<br>RO/V | **Mod-PHY Data Lane Suspend Well Power Gating Exit Ack (MDLSWPGXA):** This fields reflects the effective 'Logical AND' of mod-PHY Data Lane Suspend Well Power Gating Exit Ack for all the data lanes owned by this port.<br>• A '1b' indicates that at all the Data Lane Suspend Well Power Gating Exit Ack is asserted.<br>• A '0b' indicates that at least one of the Data Lane Suspend Well Power Gating Exit Ack has been de-asserted.<br>This bit is intended to be used for RTD3, as part of the flow to power gate the mod-PHY data lane Suspend Well. |
| 6 | 0x0<br>RO/V | **Mod-PHY Data Lane Suspend Well Power Gating Entry Ack (MDLSWPGEA):** This fields reflects the effective 'Logical OR' of mod-PHY Data Lane Suspend Well Power Gating Entry Ack for all the data lanes owned by this port.<br>• A '1b' indicates that at least one of the Data Lane Suspend Well Power Gating Entry Ack is still asserted.<br>• A '0b' indicates that all the Data Lane Suspend Well Power Ack has been de-asserted.<br>This bit is intended to be used for RTD3, as part of the flow to power gate the mod-PHY data lane Suspend Well. |
| 5 | 0x1<br>RW | **Mod-PHY Data Lane Suspend Well Power Request (MDLSWPR):** Software can write a '0b' to this field to de-assert 'Suspend Well Power Request' to all the data lanes owned by this port only when MDLSWPGXA is '1b'.<br>Software can write a '1b' to this field to assert Suspend Well Power Request to all the data lanes owned by this port only when MDLSWPGEA is '0b'.<br>This bit is intended to be used for RTD3, as part of the flow to power gate the mod-PHY data lane Suspend Well. |
| 4 | 0x0<br>RW | **PME_TO_ACK Timeout Policy (PME_TO_ACK_TOP):** When set, Root Port will trigger an exit if the link is in TXL0s or L1 and PME_TO_ACK timer times out. |
| 3:2 | 0x0<br>RW | **PME_TO_ACK Timeout (PME_TO_ACK_TO):** This timer is activated from the time the PME_Turn_Off message is sent to the link.<br>It will be reset once the timer expires or when PME_TO_ACK has been received from the downstream device and link has been put into L2/L3 Ready state.<br>Once the timer expires, the Sx sequence will continue to take the link down.<br>• 00: 1 ms (default)<br>• 01: 10 ms<br>• 10: 1 us (Pre-Si)<br>• 11: Disable |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/V/P | **L23Rdy to Detect Transition (L23RDY2DT):** When set by software, the LTSSM moves from L2/L3 Ready to Detect state.<br>This bit is cleared by hardware once the LTSSM is in Detect state.<br>The PHY power and clocks needed to enable this transition is brought up in response to this bit being set by software.<br>If the link is not in L2/L3 Ready when this bit is set, the bit is cleared immediately by hardware. |
| 0 | 0x0<br>RW/V/P | **L23Rdy Entry Request (L23RDYER):** When set by software, the corresponding Root Port will initiate the sequence to put the link into L2/L3 Ready state.<br>PME_Turn_Off message will be sent and the corresponding PME_TO_Ack response will be returned by the device.<br>Once the link enters L2/L3 Ready, this bit will be cleared by hardware.<br>Alternatively, this bit will be cleared by hardware when PME_Turn_Off timer time-out. |

## 48.2.60 PMCSRS—Offset B0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B0h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/P | **PME Status (PMESTS):** This field is utilized for Software to restore the PME Status once exit Sx State. Hardware will reflect the value to the associated PMCSR.PMESTS if this bit was written by Software. |
| 14:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 48.2.61 PPBIFCTL—Offset C0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V/P | **Capability Lock (CL):** Lock key bit for all RW-L bits of PPBIFCTL register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking.<br>Once this bit is set to a 1b from IOSF-SB, this key bit can not be unlocked.<br>Writing a 0b has no affect on this bit. |
| 30:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RO/V/P | **Bifurcation Control 0 (BIFCTL0):**<br>• 111-101: Reserved<br>• 100: x8<br>• 011: x4x4<br>• 010: x4x2x2<br>• 001: x2x2x4<br>• 000: x2x2x2x2<br>A write of '1' on the CL will lock this register field down as bifurcation control status information. |

## 48.2.62 SMICSR—Offset C8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18 | 0x0<br>RW/1C/V | **Hot Plug DLLSC SMI Status (HPLSCSMISTS):** This bit is set when rising edge detected as result of both SLOTSTS.DLLSCS and SMICSR.HPSMIE being set. |
| 17 | 0x0<br>RW/1C/V | **Hot Plug SMI Status (HPSMISTS):** This bit is set when rising edge detected as result of both SLOTSTS.PDCS and SMICSR.HPSMIE being set. |
| 16 | 0x0<br>RW/1C/V | **Power Management SMI Status (PMSMISTS):** This bit is set when rising edge detected as result of both ROOTSTS.PMESTS and SMICSR.PMSMIE being set. |
| 15:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Hot Plug SMI Enable (HPSMIE):** Enables the Root Port/DSP to generate SMI whenever a Hot Plug event is detected. |
| 0 | 0x0<br>RW | **Power Management SMI Enable (PMSMIE):** Enables the Root Port to generate SMI whenever a power management event is detected. |

## 48.2.63 SCICSR—Offset CCh

The Tag field is being used to differentiate between PMSCI (Tag = '001') and HPSCI (Tag = '000') for PMC whenever a single set of Assert/Deassert SCI is sent.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **RSVD0:** Reserved |
| 18 | 0x0<br>RW/1C/V | **Hot Plug DLLSC SCI Status (HPLSCSCISTS):** This bit is set when rising edge detected as result of both SLOTSTS.DLLSCS and SCICSR.HPSCIE being set. |
| 17 | 0x0<br>RW/1C/V | **Hot Plug SCI Status (HPSCISTS):** This bit is set when rising edge detected as result of both SLOTSTS.PDCS and SCICSR.HPSCIE being set. |
| 16 | 0x0<br>RW/1C/V | **Power Management SCI Status (PMSCISTS):** This bit is set when rising edge detected as result of both ROOTSTS.PMESTS and SCICSR.PMSCIE being set. |
| 15:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Hot Plug SCI Enable (HPSCIE):** Enables the Root Port/DSP to generate SCI whenever a Hot Plug event is detected. |
| 0 | 0x0<br>RW | **Power Management SCI Enable (PMSCIE):** Enables the Root Port to generate SCI whenever a power management event is detected. |

## 48.2.64 SSCTL—Offset D0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + D0h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RW/1C/V/P | **Stop and Scream Event Status (SSES):** This bit indicates that a stop and scream event has occurred on the Tx side of the PCI Express port due to a header or data parity error. |
| 7:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Stop and Scream LTSSM Operation (SSLTOP):**<br>• 0: PCIe will set the affected link to 'link.disabled' state when the stop and scream condition is signaled by the Transaction Layer due to poisoned data.<br>• 1: PCIe will set the affected link to 'link.disabled' state and then provide feedback to the Transaction Layer.<br>The transaction layer will then initiate a link retraining sequence via the LTSSM to bring it back alive.<br>Sometime the PCI Express port will be unable to return to normal operation due to the nature of the error that initiated the stop and scream mechanism.<br>If Software desires to retrain the link when this field is set to 0, it can perform either of the following tasks:<br>Change the LINKCTL.RL register field or Change mode field (SSLTOP) in this register to 1b or Initialize the BCTL.SBR to clear the sub-hierarchy of the PCI-Express port and reset.<br>This SBR method is only applicable to standard PCI-Express ports and not the DMI This field will only return the PCI Express link back to normal operation after a logged Stop and Scream event.<br>Errors such as header/control parity error which utilize the Stop and Scream mechanism will not be controlled by this register field. A platform reset will be required to re-enable the link.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW | **Stop and Scream Enable (SSEN):**<br>• 0: Disable Stop and Scream (normal/default operation)<br>• 1: Enables the link to perform Stop and Scream functionality.<br>When this bit is set, if PCIe root port or downstream port encounters a poisoned data traversing the outbound path, the cluster stops the data, sets the link_down condition internally and places the link in disable.<br>Inbound/outbound paths are blocked and all outstanding requests will be master aborted including Memory, I/O and configuration requests.<br><br>**Power Well:** PRST |

## 48.2.65 EXPPTMBARCTL—Offset D2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + D2h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RW/L | **Disable MBAR mPhy (DMBARMPHY):** When set to a 1b, memory transaction targeting the 64K-128K space of the EXPPTMBAR dedicated for mPhy will be prevented from accessing control registers. Memory read transactions will return 0's. Memory writes will have no effect.<br><br>**Power Well:** PRST |
| 7:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3 | 0x0<br>RW/L | **Disable Forth 4K Page (D4THPAGE):** When set to a 1b, memory transaction targeting the forth 4K page of the EXPPTMBAR will be prevented from accessing control registers.<br>Memory read transactions will return 0's. Memory writes will have no effect.<br><br>**Power Well:** PRST |
| 2 | 0x0<br>RW/L | **Disable Third 4K Page (D3RDPAGE):** When set to a 1b, memory transaction targeting the third 4K page of the EXPPTMBAR will be prevented from accessing control registers.<br>Memory read transactions return 0's. Memory writes will have no effect.<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **Disable Second 4K Page (D2NDPAGE):** When set to a 1b, memory transaction targeting the Second 4K page of the EXPPTMBAR will be prevented from accessing control registers.<br>Memory read transactions will return 0's. Memory writes will have no effect.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **Disable First 4K Page (D1STPAGE):** When set to a 1b, memory transaction targeting the first 4K page of the EXPPTMBAR will be prevented from accessing control registers.<br>Memory read transactions will return 0's. Memory writes will have no effect.<br><br>**Power Well:** PRST |

## 48.2.66 PPD0—Offset D4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits of PPD0 register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked.<br>Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 30:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13 | 0x0<br>RW/V/L | **Configuration Retry Enable (CRE):** This bit controls the ability of the external side of the NTB to respond to transactions (normal, UR, CRS).<br>The PCIe link is expected to be active after training in about 10-100ms from platform reset deassertion.<br>However, the internal controller may take longer to boot, and to initialize its registers.<br>To allow longer boot time, the CRE bit is used to ensure that the Internal side and related functionality is up and operational before allowing transactions from the external side to be processed even though link training could have been completed prior.<br>• 0: External Link handles PCIe transactions (configuration, memory, messages) in the normal fashion during linkup state.<br>• 1: External Link presents Configuration Retry Status (CRS) to the incoming configuration cycle on the external cycle in the linkup state.<br>• All other requests will be URed.<br>The NTB will respond with Configuration Retry Status (CRS) on the PCIe link for configuration cycles until such time when the internal firmware completes initialization of all internal resources.<br>If the external side sends any other request (memory, messages) before this initialization is done, it will be master aborted (UR) on the external side.<br>It is the responsibility of software to adhere to this flow.<br>This bit is reserved in Root Ports, Downstream Port and DMI EP in RC. This bit is not functional. |
| 12:10 | 0x0<br>RO | **RSVD1:** Reserved |
| 9:8 | 0x0<br>RO | **NTB Port Definition (NPD):** Value indicating the value to be loaded into the DID register (offset 02h).<br>• 00b - RP0<br>• 01b - 2 NTBs connected back to back<br>• 10b - Reserved<br>• 11b - Reserved (NTB connected to a RP)<br>The default value is determined by soft straps/fuse proxy. (DSP and NTB only) |
| 7:4 | 0x0<br>RO | **RSVD2:** Reserved |
| 3 | 0x0<br>RW/V | **Initiate Link Training 0 (ILINKTRN0):** This bit controls PCI Express port link training for associated ports.<br>A value of '1' initiates link training on ports 0, 1, 2, and 3. After writing this bit to a a '1', software can poll the Data Link Layer link active bit in the LNKSTS register to determine if a port is up and running.<br>The associated ports will not automatically initiate link training after reset unless soft strap default values have already set this bit to '1'. A write of '0' has no effect. A write of '1' will lock this register bit down and initiate link training.<br>• 0: The associated PCI Express ports have not initiated link training.<br>• 1: The associated PCI Express ports are initiating link training or have trained.<br>The default value is determined by soft straps This will reset on DL_Down events when in NTB mode<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2:1 | 0x0<br>RO | **RSVD3:** Reserved |
| 0 | 0x0<br>RW/P | **Clear ILINKTRN0 on Link down (ClrTrnEnOnLnkDn):** when set to 1, a link down event will cause the link train enable field of this register to be cleared and block subsequent training until that train enable is set again.<br>If 0, a link down will not clear the train enable unless a link down causes a reset. |

## 48.2.67 PBTXNCTL—Offset D8h

This register is reserved for the design team. These bits will control implementation specific features in the transaction layer.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **Disable APIC EOI Message (DEOIM):** This bit when set to 1b will disable downstream forwarding of the End of Interrupt (EOI) message by the Root Ports or DSP.<br>The EOI message is silently dropped. |
| 30 | 0x0<br>RW/P | **GPE Enable (GPEE):**<br>• 1b: Enables 'Assert GPE' (Deassert_GPE) messages to be sent from Root Port.<br>• 0b: Disables 'Assert GPE' (Deassert_GPE) messages.<br>This bit has an overriding effect to generate an ACPI HP event over traditional interrupts.<br>This bit only control the ports internally generated Assert, Deassert_GPE messages |
| 29:26 | 0x0<br>RW/P | **Virtual Pin Port Number (VPPN):** This field is used to identify the SMBus device and port number of the 9555 SMBus components.<br>[29:27] = SMBus Address [26] = IO Port |
| 25 | 0x0<br>RW/P | **Enable Virtual Pin Port (DVPP):** This bit when set to 1b will enable the VPP pins for the given PCI Express port.<br>• 1b: VPP is enable for this PCI Express port.<br>• 0b: VPP is disable for this PCI Express port. |
| 24 | 0x0<br>RW/P | **Hot Plug Form Factor (HPFF):** This bit determines the what hot plug form-factor a particular port.<br>• 0: CEM/Cable<br>• 1: SIOM<br>This bit is used to interpret bit 6 in the VPP serial stream for the port as either MRL# (CEM/Cable) or EMILS (SIOM) input. |
| 23:18 | 0x0<br>RW/P | **Undefined:** Design Specific Control bits |
| 17 | 0x0<br>RW/P | **Select Override Enable (SOE):** This bit when 1b enables the VPP Select Override field to program the 9555 IO pin direction. |
| 16 | 0x0<br>RW/P | **Inversion Override Enable (IOE):** This bit when 1b enables the VPP Inversion Override field to program the 9555 IO pin polarity. |
| 15:8 | 0x0<br>RW/P | **VPP Select Override (VPPSELOV):** This field is used by the HP SMBus controller to program the 9555 IO pin direction.<br>Each bit corresponds to a pin for the VPP.<br>• 0b: indicates input<br>• 1b: indicates output pin |
| 7:0 | 0x0<br>RW/P | **VPP Inversion Override (VPPIOV):** This field is used by the HP SMBus controller to program the 9555 IO pin polarity.<br>Each bit corresponds to a pin for the VPP.<br>• 1b: Invert Polarity.<br>• 0b: Polarity not inverted. |

## 48.2.68 PBPHYCTL—Offset E0h

This register is reserved for the design team. These bits will control implementation specific features in the physical layer.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + E0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/P | **Undefined:** Design Specific Control bits |

## 48.2.69 PLKCTL—Offset EAh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + EAh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW/L | **Lnk Layer Capability Lock (LLCL):** Lock key bit for all RWS-L bits located in the following CSRs. XPLDEF XPLDEF2 LLRBERRINJCTL RXPERRINJCTL CRCERRINJCTL CRCERRINJMSK RTRYCTL PACKTHRESH<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **Txn Layer Capability Lock (TLCL):** Lock key bit for all RWS-L bits located in XPEINJ[1:0], XPCDTHROTTLEP,N,C, XPTDEF*, MBAR01CTL EXPPTMBARCTL.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RWS-L bits (capabilities, next capability pointer, SSID/SVID, slot register, etc) bits for the function.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |

## 48.2.70 LTRCSR—Offset ECh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + ECh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits of LTRCSR and LTROVR register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |
| 30:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RW/1C/V | **Invalid LTR Latency Scale Value Received (IVLSVR):** Set to '1' by hardware when an LTR message is received with Not Permitted Latency Scale Value and both the LTR Mechanism Supported bit and the LTR Mechanism Enable bit are set.<br>Cleared to '0' by software writing 1 to this register. |
| 16 | 0x0<br>RW/1C/V | **LTR Received (LTRR):** Set to '1' by hardware when an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable bit are set.<br>It can be set/cleared by hardware internally, or software by writing 1 to clear it. |
| 15:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3 | 0x0<br>RW/P | **LTR Aggregation Mode Disable (LTRAMD):** Default is 0 to enable LTR aggregation mode.<br>Set 1 to disable LTR Aggregation Mode. |
| 2 | 0x0<br>RW/L | **LTR Override Policy (LTROVRPLCY):** This register bit defines the LTR override behavior when the respective LTR Override Enable bits are set.<br>• 0: The LTR Override values will be in effect when the link is up and in D0, until a valid new LTR message is received from the device.<br>• 1: The LTR Override values will be in effect when the link is up and in D0. LTR values from the valid new LTR message received from device will be ignored.<br>***Note:*** This register should be programmed prior to enabling LTR override and should not be modified after LTR override has been enabled.<br><br>**Power Well:** PRST |
| 1 | 0x0<br>RW/L | **LTR No-Snoop Override Enable (LTRNSOVREN):** When this bit is set, the latency tolerance values for this port will be overridden to the values programmed in the LTRNSLOVRV, LTRNSLSOVRV and LTRNSROVR fields.<br>The latency values from the LTR messages received from the devices will be ignored.<br>Hardware should treat this bit as dynamic and the transition of this bit from '0' to '1' will cause a new IOSF sideband LTR message to be sent to update PMC if the new aggregated latency values change as a result of this.<br><br>**Power Well:** PRST |
| 0 | 0x0<br>RW/L | **LTR Snoop Override Enable (LTRSOVREN):** When this bit is set, the latency tolerance values for this port will be overridden to the values programmed in the LTRSLOVRV, LTRSLSOVRV and LTRSROVR fields.<br>The latency values from the LTR messages received from the devices will be ignored.<br>Hardware should treat this bit as dynamic and the transition of this bit from '0' to '1' will cause a new IOSF sideband LTR message to be sent to update PMC if the new aggregated latency values change as a result of this.<br><br>**Power Well:** PRST |

## 48.2.71 LTRL—Offset F0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + F0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO/V | **LTR No-Snoop Latency Field (LTRNSLF):** This field contains the No-Snoop Latency field of the latest received LTR Message.<br>When an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable field are set, bytes 12:13 of the PCIe header are stored in this register.<br>This includes the case where the Latency Scale Value contains 'Not Permitted' encoding. |
| 15:0 | 0x0<br>RO/V | **LTR Snoop Latency Field (LTRSLF):** This field contains the Snoop Latency field of the latest received LTR Message.<br>When an LTR message is received and both the LTR Mechanism Supported bit and LTR Mechanism Enable field are set, bytes 14:15 of the PCIe header are stored in this register.<br>This includes the case where the Latency Scale Value contains 'Not Permitted' encoding. |

## 48.2.72 LTROVR—Offset F4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + F4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **LTR No-Snoop Requirement Bit Override (LTRNSROVR):** This field contains the No-Snoop Requirement bit override value for this particular PCIe root port.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 30:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:26 | 0x0<br>RW/L | **LTR No-Snoop Latency Scale Override Value (LTRNSLSOVRV):** This field contains the Snoop latency scale override value for this particular PCIe root port.<br>• 000: LTRNSLOVRV times 1 ns<br>• 001: LTRNSLOVRV times 32 ns<br>• 010: LTRNSLOVRV times 1,024 ns<br>• 011: LTRNSLOVRV times 32,768 ns<br>• 100: LTRNSLOVRV times 1,048,576 ns<br>• 101: LTRNSLOVRV times 33,554,432 ns<br>• Others: Not Permitted.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 25:16 | 0x0<br>RW/L | **LTR No-Snoop Latency Override Value (LTRNSLOVRV):** This field contains the No-Snoop latency override value for this particular PCIe root port.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 15 | 0x0<br>RW/L | **LTR Snoop Requirement Bit Override (LTRSROVR):** This field contains the No-Snoop Requirement bit override value for this particular PCIe root port.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 14:13 | 0x0<br>RO | **RSVD1:** Reserved |
| 12:10 | 0x0<br>RW/L | **LTR Snoop Latency Scale Override Value (LTRSLSOVRV):** This field contains the Snoop latency scale override value for this particular PCIe root port.<br>• 000: LTRSLOVRV times 1 ns<br>• 001: LTRSLOVRV times 32 ns<br>• 010: LTRSLOVRV times 1,024 ns<br>• 011: LTRSLOVRV times 32,768 ns<br>• 100: LTRSLOVRV times 1,048,576 ns<br>• 101: LTRSLOVRV times 33,554,432 ns<br>• Others: Not Permitted.<br>This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |
| 9:0 | 0x0<br>RW/L | **LTR Snoop Latency Override Value (LTRSLOVRV):** This field contains the Snoop latency override value for this particular PCIe root port. This field is Read-Only when LTRCSR.CL is set.<br><br>**Power Well:** PRST |

## 48.2.73 INTXSWZCTL—Offset F8h

This register provides software the ability to swizzle the legacy interrupts (INTx) from each port and remap them to a different interrupt type for purposes of interrupt rebalancing. This swizzling only applies to inbound INTx messages that arrive on the secondary side of the switch or root port. This register does not affect PCI-Express defined INTx re-map based on device # that is required by the PCI-Express base specification.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + F8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RW/L | **INTx Swizzle (INTXSWZ):** The encoding below defines the target INTx type to which the incoming INTx message is mapped to for that port.<br>• 000b:INTA=)INTA, INTB=)INTB, INTC=)INTC, INTD=)INTD,<br>• 001b:INTA=)INTB, INTB=)INTC, INTC=)INTD, INTD=)INTA,<br>• 010b:INTA=)INTC, INTB=)INTD, INTC=)INTA, INTD=)INTB,<br>• 011b:INTA=)INTD, INTB=)INTA, INTC=)INTB, INTD=)INTC,<br>• 100b:INTA=)INTA, INTB=)INTA, INTC=)INTA, INTD=)INTA,<br><br>**Power Well:** PRST |

## 48.2.74 CFGAGTERR—Offset FCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + FCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RW/1C/V/P | **Data Parity Error Status (DPES):** Reserved. |
| 9 | 0x0<br>RO | **RSVD1:** Reserved |
| 8 | 0x0<br>RW/1C/V/P | **Transaction Layer Internal Parity Error Status (TLIPES):** Reserved. |
| 7 | 0x0<br>RW/1C/V/P | **Link Layer Internal Parity Error Status (LLIPES):** Reserved. |
| 6 | 0x0<br>RW/1C/V/P | **Physical Layer Internal Parity Error Status (PLIPES):** Reserved. |
| 5 | 0x0<br>RO | **RSVD2:** Reserved |
| 4 | 0x0<br>RW/P | **Data Parity Error Mask (DPEM):** Reserved. |
| 3 | 0x0<br>RO | **RSVD3:** Reserved |
| 2 | 0x0<br>RW/P | **Transaction Layer Internal Parity Error Mask (TLIPEM):** Reserved. |
| 1 | 0x0<br>RW/P | **Link Layer Internal Parity Error Mask (LLIPEM):** Reserved. |
| 0 | 0x0<br>RW/P | **Physical Layer Internal Parity Error Mask (PLIPEM):** Reserved. |

## 48.2.75 AERCAPHDR—Offset 100h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 100h

**Default:** 13810001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x138<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities list. (ACSCAPHDR).<br>Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0x1<br>RO | **Extended Capability ID (EXCAPID):** Identifies the function as Advanced Error Reporting capable. |

## 48.2.76 ERRUNCSTS—Offset 104h

This register reports the error status of individual uncorrectable error sources. An individual error status bit that is set to 1 indicates that a particular error occurred. Software can clear an error status by writing a 1 to the respective bit.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RSVD0:** Reserved |
| 26 | 0x0<br>RW/1C/V/P | **Poisoned TLP Egress Blocked Status (PTLPEBS):** Reserved. |
| 25 | 0x0<br>RO | **TLP Prefix Blocked Error Status (TPBES):** Reserved. |
| 24 | 0x0<br>RW/1C/V/P | **Atomic Egress Blocked Status (AEBE):** This bit is set whenever an Atomic OP TLP is blocked on any egress port |
| 23 | 0x0<br>RW/1C/V/P | **MC Blocked TLP Status (MCE):** This bit is set whenever a Multicast TLP is blocked. |
| 22 | 0x0<br>RW/1C/V/P | **Uncorrectable Internal Error Status (UIE):** This bit is set whenever an uncorrectable internal error is detected. |
| 21 | 0x0<br>RW/1C/V/P | **ACS Violation Status (ACSE):** This bit is set whenever an ACS violation is detected by the PCI Express port. |
| 20 | 0x0<br>RW/1C/V/P | **Unsupported Request Error Status (URE):** This bit is set whenever an unsupported request is detected on PCI Express port |
| 19 | 0x0<br>RW/1C/V/P | **ECRC Error Status (ECRCE):** ECRC checking is optional for project specific. |
| 18 | 0x0<br>RW/1C/V/P | **Malformed TLP Status (MTLPE):** This bit is set when it receives a malformed TLP. Header logging is performed. |
| 17 | 0x0<br>RW/1C/V/P | **Receiver Overflow Status (ROE):** This bit is set when the PCI Express interface unit receive buffers overflow. |
| 16 | 0x0<br>RW/1C/V/P | **Unexpected Completion Status (UCE):** This bit is set whenever a completion is received with a requestor ID that does not match side A or side B, or when a completion is received with a matching requestor ID but an unexpected tag field. Header logging is performed. |
| 15 | 0x0<br>RW/1C/V/P | **Completer Abort Status (CAE):** The bridge sets this bit and logs the header associated with the request when the configuration unit signals a completer abort. |
| 14 | 0x0<br>RW/1C/V/P | **Completion Timeout Status (CTE):** For Switch Ports, this bit is set if the Switch Port issues Non-Posted Requests on its own behalf (vs. only forwarding such as Requests generated by other devices). |
| 13 | 0x0<br>RW/1C/V/P | **Flow Control Protocol Error Status (FCE):** This bit is set when a flow control protocol error is detected. |
| 12 | 0x0<br>RW/1C/V/P | **Poisoned TLP Received Status (PTLPE):** This bit is set and the bridge logs the header when a poisoned TLP is received from PCI Express. |
| 11:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5 | 0x0<br>RW/1C/V/P | **Surprise Down Error Status (SLDE):** This bit is set when a surprise down error is detected. This bit does not apply to the upstream port. |
| 4 | 0x0<br>RW/1C/V/P | **Data Link Protocol Error Status (DLPE):** This bit is set when a data link protocol error is detected. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 48.2.77 ERRUNCMSK—Offset 108h

This register controls the reporting of individual uncorrectable errors by device to the host bridge via a PCI Express error message. This register also controls the logging of the header. Refer to the PCI Express specifications for details of how the mask bits function. A masked error (respective bit set in the mask register) is not reported to the host bridge by the switch, nor is the header logged (status bits unaffected by the mask bit). There is a mask bit per bit of the Uncorrectable Error Status Register. The attr. of AER Error Mask bits in VRSP will be changed to be compliant with PCIe Spec.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 108h

**Default:** 04400000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RSVD0:** Reserved |
| 26 | 0x1<br>RW/P | **Poisoned TLP Egress Blocked Mask (PTLPEBM):** Reserved. |
| 25 | 0x0<br>RO | **TLP Prefix Blocked Error Mask (TPBES):** Reserved. |
| 24 | 0x0<br>RW/P | **AtomicOp Egress Blocked Mask (AEBEM):** Reserved. |
| 23 | 0x0<br>RW/P | **MC Blocked TLP Mask (MCEM):** Reserved. |
| 22 | 0x1<br>RW/P | **Uncorrectable Internal Error Mask (UIEM):** Reserved. |
| 21 | 0x0<br>RW/P | **ACS Violation Mask (ACSEM):** Reserved. |
| 20 | 0x0<br>RW/P | **Unsupported Request Error Mask (UREM):** Reserved. |
| 19 | 0x0<br>RW/P | **ECRC Error Mask (ECRCEM):** Reserved. |
| 18 | 0x0<br>RW/P | **Malformed TLP Mask (MTLPEM):** Reserved. |
| 17 | 0x0<br>RW/P | **Receiver Overflow Mask (ROEM):** Reserved. |
| 16 | 0x0<br>RW/P | **Unexpected Completion Mask (UCEM):** Reserved. |
| 15 | 0x0<br>RW/P | **Completer Abort Mask (CAEM):** Reserved. |
| 14 | 0x0<br>RW/P | **Completion Timeout Mask (CTEM):** Reserved. |
| 13 | 0x0<br>RW/P | **Flow Control Protocol Error Mask (FCEM):** Reserved. |
| 12 | 0x0<br>RW/P | **Poisoned TLP Received Mask (PTLPEM):** Reserved. |
| 11:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5 | 0x0<br>RW/P | **Surprise Down Error Mask (SLDEM):** This bit does not apply to the upstream port or End point |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW/P | **Data Link Protocol Error Mask (DLPEM):** Reserved. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 48.2.78 ERRUNCSEV—Offset 10Ch

This register controls whether an individual uncorrectable error is reported as a fatal error. An uncorrectable error is reported as fatal when the corresponding error bit in this register is set. When the bit is cleared, the corresponding error is considered non-fatal. The attr. of AER Error Severity bits in VRSP will be changed to be compliant with PCIe Spec.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 10Ch

**Default:** 00462030h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RSVD0:** Reserved |
| 26 | 0x0<br>RW/P | **Poisoned TLP Egress Blocked Severity (PTLPEBS):** Reserved. |
| 25 | 0x0<br>RO | **TLP Prefix Blocked Error Severity (TPBES):** Reserved. |
| 24 | 0x0<br>RW/P | **AtomicOp Egress Blocked Severity (AEBES):** Reserved. |
| 23 | 0x0<br>RW/P | **MC Blocked TLP Severity (MCES):** Reserved. |
| 22 | 0x1<br>RW/P | **Uncorrectable Internal Error Severity (UIES):** Reserved. |
| 21 | 0x0<br>RW/P | **ACS Violation Severity (ACSES):** Reserved. |
| 20 | 0x0<br>RW/P | **Unsupported Request Error Severity (URES):** Reserved. |
| 19 | 0x0<br>RW/P | **ECRC Error Severity (ECRCES):** Reserved. |
| 18 | 0x1<br>RW/P | **Malformed TLP Severity (MTLPES):** Reserved. |
| 17 | 0x1<br>RW/P | **Receiver Overflow Error Severity (ROES):** Reserved. |
| 16 | 0x0<br>RW/P | **Unexpected Completion Error Severity (UCES):** Reserved. |
| 15 | 0x0<br>RW/P | **Completer Abort Error Severity (CAES):** Reserved. |
| 14 | 0x0<br>RW/P | **Completion Timeout Error Severity (CTES):** Reserved. |
| 13 | 0x1<br>RW/P | **Flow Control Protocol Error Severity (FCES):** Reserved. |
| 12 | 0x0<br>RW/P | **Poisoned TLP Received Severity (PTLPES):** Reserved. |
| 11:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5 | 0x1<br>RW/P | **Surprise Down Error Severity (SLDES):** This bit does not apply to the upstream port or end point |
| 4 | 0x1<br>RW/P | **Data Link Protocol Error Severity (DLPES):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 48.2.79 ERRCORSTS—Offset 110h

This register reports the error status of individual correctable error sources on a PCI Express device. An individual error status bit set to 1 indicates that a particular error has occurred. Software can clear the error status by writing a 1 to the respective bit.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 110h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/1C/V/P | **Header Log Overflow Error (HLOE):** Reserved. |
| 14 | 0x0<br>RW/1C/V/P | **Correctable Internal Error (CIE):** Reserved. |
| 13 | 0x0<br>RW/1C/V/P | **Advisory Non-Fatal Error (ANFE):** Reserved. |
| 12 | 0x0<br>RW/1C/V/P | **Replay Timer Timeout Error (RTTE):** The PCIe sets this bit when replay timer time-out occurs. |
| 11:9 | 0x0<br>RO | **RSVD1:** Reserved |
| 8 | 0x0<br>RW/1C/V/P | **Replay Number Rollover Error (RNRE):** The PCIe sets this bit when the replay number rolls over from 11 to 00. |
| 7 | 0x0<br>RW/1C/V/P | **Bad DLLP Error (BDLLPE):** The switch sets this bit on CRC errors on DLLP. |
| 6 | 0x0<br>RW/1C/V/P | **Bad TLP Error (BTLPE):** The switch sets this bit on CRC errors on TLP. |
| 5:1 | 0x0<br>RO | **RSVD2:** Reserved |
| 0 | 0x0<br>RW/1C/V/P | **Receiver Error (RE):** The PCIe sets this bit when the physical layer detects a receiver error. |

## 48.2.80 ERRCORMSK—Offset 114h

This register controls the reporting of individual correctable errors via ERR_COR message. A masked error (respective bit set in mask register) is not reported to the host bridge by the switch. There is a mask bit corresponding to every bit in the Correctable Error Status Register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 114h

**Default:** 0000E000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x1<br>RW/P | **Header Log Overflow Error Mask (HLOEM):** Reserved. |
| 14 | 0x1<br>RW/P | **Correctable Internal Error Mask (CIEM):** Reserved. |
| 13 | 0x1<br>RW/P | **Advisory Non-Fatal Error Mask (ANFEM):** Reserved. |
| 12 | 0x0<br>RW/P | **Replay Timer Timeout Error Mask (RTTEM):** Reserved. |
| 11:9 | 0x0<br>RO | **RSVD1:** Reserved |
| 8 | 0x0<br>RW/P | **Replay Number Rollover Error Mask (RNREM):** Reserved. |
| 7 | 0x0<br>RW/P | **Bad DLLP Error Mask (BDLLPEM):** Reserved. |
| 6 | 0x0<br>RW/P | **Bad TLP Error Mask (BTLPEM):** Reserved. |
| 5:1 | 0x0<br>RO | **RSVD2:** Reserved |
| 0 | 0x0<br>RW/P | **Receiver Error Mask (REM):** Reserved. |

## 48.2.81 AERCAPCTL—Offset 118h

This register gives the status and control for ECRC checks and also the pointer to the first uncorrectable error that happened.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 118h

**Default:** 00001000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:13 | 0x0<br>RO | **RSVD0:** Reserved |
| 12 | 0x1<br>RW/L | **Completion Timeout Prefix/Header Log Capable (CTPHLC):** If Set, this bit indicates that the Function records the prefix/header of Request TLPs that experience a Completion Timeout error.<br><br>**Power Well:** PRST |
| 11 | 0x0<br>RO/V/P | **TLP Prefix Log Present (TLPPLP):** If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information.<br>If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. |
| 10 | 0x0<br>RO | **Multiple Header Recording Enable (MHRE):** When set, this bit enables the function to record more than one error header. |
| 9 | 0x0<br>RO | **Multiple Header Recording Capable (MHRC):** Supports more than one error header. |
| 8 | 0x0<br>RW/P | **ECRC Check Enable (ECE):** When Set, ECRC checking is enabled.<br>Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b. |
| 7 | 0x0<br>RW/L | **ECRC Check Capable (ECC):** If Set, this bit indicates that the Function is capable of checking ECRC.<br><br>**Power Well:** PRST |
| 6 | 0x0<br>RW/P | **ECRC Generation Enable (EGE):** When Set, ECRC generation is enabled.<br>Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b.<br>Default value of this bit is 0b. |
| 5 | 0x0<br>RW/L | **ECRC Generation Capable (EGC):** If Set, this bit indicates that the Function is capable of generating ECRC.<br><br>**Power Well:** PRST |
| 4:0 | 0x0<br>RO/V/P | **First Error Pointer (FEP):** This field identifies the bit position of the first error reported in the Uncorrectable Error Status Register (xref).<br>This register re-arms itself (which does not change its current value) as soon as the error status bit indicated by the pointer is cleared by the software by writing a 1 to that status bit. |

## 48.2.82 AERHDRLOG1—Offset 11Ch

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 11Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 48.2.83 AERHDRLOG2—Offset 120h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 48.2.84 AERHDRLOG3—Offset 124h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 48.2.85 AERHDRLOG4—Offset 128h

This register logs the 4 Dwords of the transaction header for PCI Express errors.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V/P | **TLP Header Log (TLPHDRLOG):** As soon as an error is logged in this register, it remains locked for further error-logging until the software clears the status bit that caused the header log (in other words, until the error pointer is re-armed for logging again). |

## 48.2.86 ROOTERRCMD—Offset 12Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 12Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **Fatal Error Reporting Enable (FERE):** When set, this bit enables the generation of an interrupt when a Fatal error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 1 | 0x0<br>RW | **Non-Fatal Error Reporting Enable (NFERE):** When set, this bit enables the generation of an interrupt when a Non-Fatal error is reported by any of the functions in the hierarchy associated with this Root Port. |
| 0 | 0x0<br>RW | **Correctable Error Reporting Enable (CERE):** When set, this bit enables the generation of an interrupt when a Correctable error is reported by any of the functions in the hierarchy associated with this Root Port. |

## 48.2.87 ROOTERRSTS—Offset 130h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO/V | **Advanced Error Interrupt Message Number (AEMN):** This register indicates which MSI/MSI-X vector is used for the interrupt message generated in association with any of the status bits of this Capability.<br>For MSI, the value in this register indicates the offset between the base Message Data and the interrupt message that is generated.<br>Hardware is required to update this field so that it is correct if the number of MSI Messages assigned to the Function changes when software writes to the Multiple Message Enable field in the MSI Message Control register.<br>For MSI-X, the value in this register indicates which MSI-X Table entry is used to generate the interrupt message.<br>The entry must be one of the first 32 entries even if the Function implements more than 32 entries.<br>For a given MSI-X implementation, the entry must remain constant.<br>For remaining fields info, please refer to PCIe Spec. for more details. |
| 26:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW/1C/V/P | **Fatal Error Message Received (FEMR):** Reserved. |
| 5 | 0x0<br>RW/1C/V/P | **Non-Fatal Error message Received (NFEMR):** Reserved. |
| 4 | 0x0<br>RW/1C/V/P | **First Uncorrectable Fatal (FUF):** Reserved. |
| 3 | 0x0<br>RW/1C/V/P | **Multiple Error Fatal/Non-Fatal Received (MEFR):** Reserved. |
| 2 | 0x0<br>RW/1C/V/P | **Error Fatal/Non-Fatal Received (EFR):** Reserved. |
| 1 | 0x0<br>RW/1C/V/P | **Multiple Error Correctable Error Received (MCER):** Reserved. |
| 0 | 0x0<br>RW/1C/V/P | **Correctable Error Received (CER):** Reserved. |

## 48.2.88 ERRSRCID—Offset 134h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 134h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO/V/P | **Fatal/Non-Fatal Error Source ID (EFSID):** Requester ID of the source when an Fatal or No Fatal Error is received and the Fatal Error Message Received (FEMR) or Non-Fatal Error message Received (NFEMR) bit is not already set. |
| 15:0 | 0x0<br>RO/V/P | **Correctable Error Source ID (ECSID):** Requester ID of the source when a correctable error is received and the Correctable Error Received (CER) bit is not already set. |

## 48.2.89 ACSCAPHDR—Offset 138h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 138h

**Default:** 1501000Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x150<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities. Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0xd<br>RO | **Extended Capability ID (ECID):** Identifies the function as Access Control Services capable. |

## 48.2.90 ACSCAP—Offset 13Ch

This register identifies the Access Control Services (ACS) capabilities.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 13Ch

**Default:** 005Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Egress Control Vector Size (ECVS):** Indicates the number of bits in the Egress Control Vector. This is set to 00h as ACS P2P Egress Control (ACSP2PEC) bit 5 in this register is 0b. |
| 7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x1<br>RO | **ACS Direct Translated P2P (TACSDTP2P):** Indicates that the component does implement ACS Direct Translated P2P. Required for Root Ports that support Address Translation Services (ATS) and also support peer-to-peer traffic with other Root Ports, required for Switch Downstream Ports, |
| 5 | 0x0<br>RO | **ACS P2P Egress Control (EACSP2PEC):** Hardwired to 0. Indicates that the component does not implement ACS P2P Egress Control. |
| 4 | 0x1<br>RO | **ACS Upstream Forwarding (UACSUF):** Indicates that the component implements ACS Upstream Forwarding. |
| 3 | 0x1<br>RO | **ACS P2P Completion Redirect (CACSP2PRR):** Indicates that the component implements ACS P2P Completion Redirect.<br>This includes Read Completions, AtomicOp Completions, and other Completions, either with or without data. |
| 2 | 0x1<br>RO | **ACS P2P Request Redirect (RACSP2PRR):** Indicates that the component implements ACS P2P Request Redirect. |
| 1 | 0x1<br>RO | **ACS Translation Blocking (BACSTB):** Indicates that the component implements ACS Translation Blocking. |
| 0 | 0x1<br>RO | **ACS Source Validation (VACSSV):** Indicates that the component implements ACS Source Validation. |

## 48.2.91 ACSCTL—Offset 13Eh

This register identifies the Access Control Services (ACS) control bits.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 13Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x0<br>RW | **ACS Direct Translated P2P Enable (TACSP2PECE):** When Set, overrides the ACS P2P Request Redirect and ACS P2P Egress Control mechanisms with peer-to-peer Memory Requests whose Address Translation (AT) field indicates a Translated address.<br>This bit is ignored if ACS Translation Blocking (B) is enabled. |
| 5 | 0x0<br>RO | **ACS P2P Egress Control Enable (EACSP2PECE):** This is hardwired to 0b as the component does not implement ACS P2P Egress Control. |
| 4 | 0x0<br>RW | **ACS Upstream Forwarding Enable (UACSUFE):** When set, the component forwards upstream any Request or Completion TLPs it receives that were redirected upstream by a component lower in the hierarchy.<br>Note that the U bit only applies to upstream TLPs arriving at a Downstream Port, and whose normal routing targets the same Downstream Port. |
| 3 | 0x0<br>RW | **ACS P2P Completion Redirect Enable (CACSP2PCRE):** Determines when the component redirects peer-to-peer Completions upstream, applicable only to Read Completions whose Relaxed Ordering Attribute is clear. |
| 2 | 0x0<br>RW | **ACS P2P Request Redirect Enable (RACSP2PRRE):** This bit determines when the component redirects peer-to-peer Requests upstream. |
| 1 | 0x0<br>RW | **ACS Translation Blocking Enable (BACSTBE):** When set, the component blocks all upstream Memory Requests whose Address Translation (AT) field is not set to the default value. |
| 0 | 0x0<br>RW | **ACS Source Validation Enable (VACSSVE):** When set, the component validates the Bus Number from the Requester ID of upstream Requests against the secondary / subordinate Bus Numbers. |

## 48.2.92 ERRUNCDETMSK—Offset 140h

A value of 1 masks the field and prevents detection as well as logging.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 140h

**Default:** 04000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RSVD0:** Reserved |
| 26 | 0x1<br>RW/P | **Poisoned TLP Egress Blocked Detect Mask (PTLPEBDM):** Reserved. |
| 25 | 0x0<br>RO | **TLP Prefix Blocked Error Detect Mask (TPBEDM):** Reserved. |
| 24 | 0x0<br>RW/P | **AtomicOp Egress Blocked Error Detect Mask (AEBEDM):** Reserved. |
| 23 | 0x0<br>RW/P | **MC Blocked TLP Error Detect Mask (MCEDM):** Reserved. |
| 22 | 0x0<br>RW/P | **Uncorrectable Internal Error Detect Mask (UIEDM):** Reserved. |
| 21 | 0x0<br>RW/P | **ACS Violation Error Detect Mask (ACSEDM):** Reserved. |
| 20 | 0x0<br>RW/P | **Unsupported Request Error Detect Mask (UREDM):** Reserved. |
| 19 | 0x0<br>RW/P | **ECRC Check Error Mask (ECRCEDM):** Supported |
| 18 | 0x0<br>RW/P | **Malformed TLP Error Detect Mask (MTLPEDM):** Reserved. |
| 17 | 0x0<br>RW/P | **Receiver Overflow Error Detect Mask (ROEDM):** Reserved. |
| 16 | 0x0<br>RW/P | **Unexpected Completion Error Detect Mask (UCEDM):** Reserved. |
| 15 | 0x0<br>RW/P | **Completer Abort Error Detect Mask (CAEDM):** Reserved. |
| 14 | 0x0<br>RW/P | **Completion Timeout Error Detect Mask (CTEDM):** Reserved. |
| 13 | 0x0<br>RW/P | **Flow Control Error Detect Mask (FCEDM):** Reserved. |
| 12 | 0x0<br>RW/P | **Poisoned TLP Error Detect Mask (PTLPEDM):** Reserved. |
| 11:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5 | 0x0<br>RW/P | **Surprise Down Error Detect Mask (SLDEDM):** Reserved. |
| 4 | 0x0<br>RW/P | **Data Link Protocol Error Detect Mask (DLPEDM):** Reserved. |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved |

## 48.2.93 ERRCORDETMSK—Offset 144h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 144h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/P | **Header Log Overflow Error Detect Mask (HLOEDM):** Reserved. |
| 14 | 0x0<br>RW/P | **Correctable Internal Error Detect Mask (CIEDM):** Reserved. |
| 13 | 0x0<br>RW/P | **Advisory Non-Fatal Error Detect Mask (ANFEDM):** Reserved. |
| 12 | 0x0<br>RW/P | **Replay Timer Timeout Error Detect Mask (RTTEDM):** Reserved. |
| 11:9 | 0x0<br>RO | **RSVD1:** Reserved |
| 8 | 0x0<br>RW/P | **Replay Number Rollover Error Detect Mask (RNREDM):** Reserved. |
| 7 | 0x0<br>RW/P | **Bad DLLP Error Detect Mask (BDLLPEDM):** Reserved. |
| 6 | 0x0<br>RW/P | **Bad TLP Error Detect Mask (BTLPEDM):** Reserved. |
| 5:1 | 0x0<br>RO | **RSVD2:** Reserved |
| 0 | 0x0<br>RW/P | **Receiver Error Detect Mask (REDM):** Reserved. |

## 48.2.94 ROOTERRDETMSK—Offset 148h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 148h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW/P | **Received Fatal Message Detect Mask (RFMDM):** Reserved. |
| 1 | 0x0<br>RW/P | **Received Non-Fatal Message Detect Mask (RNFMDM):** Reserved. |
| 0 | 0x0<br>RW/P | **Received Correctable Message Detect Mask (RCEMDM):** Reserved. |

## 48.2.95 MCSTCAPHDR—Offset 150h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 150h

**Default:** 18010012h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x180<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities. Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0x12<br>RO | **Extended Capability ID (ECID):** Identifies the function as Multicast Capability. |

## 48.2.96 MCSTCAP—Offset 154h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 154h

**Default:** 000Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **MC ECRC Regeneration supported (MCERS):**<br>1: When set, this bit indicates that ECRC regeneration is supported. |
| 14:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:0 | 0xf<br>RO | **MC Max Group (MCMG):** This field indicates the maximum number of Multicast Groups that the component supports, encoded as M-1.<br>A value of 00h indicates that one multicast Group is supported. |

## 48.2.97 MCSTCTL—Offset 156h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 156h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **MC Enable (MCEN):** When set, this bit indicates the Multicast Capability is enabled for the component.<br>This is a global field that controls MC for the specific function. |
| 14:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:0 | 0x0<br>RW | **MC Number Group (MCNG):** This field indicates the number of Multicast Groups configured for use, encoded as N-1.<br>A value of 00h indicates that one Multicast Group is configured for use.<br>Behavior is undefined if value exceeds MC Max Group.<br>This parameter indirectly defines the upper limit of the Multicast address range.<br>This field is ignored if MC Enable is Clear. |

## 48.2.98 MCSTBAR—Offset 158h

The MC_Base_Address register contains the MC_Base_Address and the MC_Index_Position for calculating whether a MC hit occurred or not. Refer to the PCI Local Bus Specification, Revision 3.0 for details on how the MC hit/miss and the MC group number is calculated. This register is 4K aligned and hence only the top 20 bits are useable for the base address.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 158h

**Default:** 0000000Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Lower Address 32-bit (L32ADR):** Reserved. |
| 11:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:0 | 0xc<br>RW | **MC Index Position (MCIP):** This field indicates the location of the LSB of the Multicast Group number within the address.<br>Behavior is undefined if this field is less than 12 when MC Enable is set. |

## 48.2.99 MCSTUBAR—Offset 15Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 15Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address 32-bit (U32ADR):** Reserved. |

## 48.2.100 MCSTRCV—Offset 160h

The MC_Receive register provides a bit vector denoting which Multicast groups the Function should accept

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 160h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW | **MC Receive (MCR):** For each bit that is set, this function gets a copy of any Multicast TLPs for the associated Multicast Group. Bits above MC Number Groups are ignored by hardware. |

## 48.2.101 MCSTBLKALL—Offset 168h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 168h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW | **MC Block All (MCBA):** For each bit that is set, this function is blocked from sending TLPs to the associated Multicast Group.<br>Bits above MC Number Groups are ignored by hardware. |

## 48.2.102 MCSTOLBAR—Offset 178h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 178h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **MC Overlay BAR (MCOVRBAR):** Specifies the base address of the window onto which MC TLPs passing through this function will be overlaid.<br>The overlay BAR is 4K aligned. Default is 0. |
| 11:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:0 | 0x0<br>RW | **MC Overlay Size (MCOVRSZ):** If 6 or greater, specifies the size in bytes of the overlay aperture as a power of 2. If less than 6, disables the overlay mechanism. Default is 0. |

## 48.2.103 MCSTUOLBAR—Offset 17Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 17Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **MC Overlay BAR Upper 32 (MCOVRBARU32):** Specifies the base address of the window onto which MC TLPs passing through this function will be overlaid. |

## 48.2.104 EINJCAPHDR—Offset 180h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 180h

**Default:** 1901000Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x190<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities. Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0xb<br>RO | **Extended Capability ID (ECID):** Vendor Defined Capability |

## 48.2.105 EINJHDR—Offset 184h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 184h

**Default:** 00A00003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0xa<br>RO | **VSEC Length (VSECLEN):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Express Extended Capability header, the Vendor-Specific header, and the Vendor-Specific registers. |
| 19:16 | 0x0<br>RO | **VSEC Rev (VSECREV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field. |
| 15:0 | 0x3<br>RO | **VSEC ID (VSECID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure.<br>Software must qualify the Vendor ID before interpreting this field.<br>3h is assigned to WHEA Error Injection. |

## 48.2.106 EINJCTL—Offset 188h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 188h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **Inject Completion Timeout Error (INJCTOERR):** When this bit is written from 0 to 1, one and only one error assertion pulse is produced on the error source signal for the given port.<br>This error will appear equivalent to an actual error assertion because this event is OR'd into the existing error reporting structure.<br>To log another error, this bit must be cleared first, before setting again.<br>Leaving this bit in a 1 state does not produce a persistent error condition. |
| 1 | 0x0<br>RW | **Inject Receiver Error (INJRCVERR):** When this bit is written from 0 to 1, one and only one error assertion pulse is produced on the error source signal for the given port.<br>This error will appear equivalent to an actual error assertion because this event is OR'd into the existing error reporting structure.<br>To log another error, this bit must be cleared first, before setting again.<br>Leaving this bit in a 1 state does not produce a persistent error condition. |
| 0 | 0x0<br>RW/L | **Error Injection Disable (EINJDIS):** This bit disables the usage of the PCIe Error Injection bits for EINJCTL.INJRCVERR, INJCTOERR This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked.<br>Writing a 0b has no affect on this bit.<br><br>**Power Well:** PRST |

## 48.2.107 SECEXPCAP—Offset 200h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 200h

**Default:** 00010019h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW/L | **Next Capability Offset (NCO):** Contains the offset of the next structure in the Extended Capabilities list.<br>Set to Null.<br>Software will need to program this field appropriately for linked capabilities list if default is not supported.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (CV):** Indicates the version of the Capability structure present. |
| 15:0 | 0x19<br>RO | **Extended Capability ID (ECID):** PCI Express Extended Capability ID for the Secondary PCI Express Extended Capability is 0019h |

## 48.2.108 LINKCTL3—Offset 204h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 204h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:9 | 0x0<br>RW | **Enable Lower SKP OS Generation Vector (ELSKPOSGV):** When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture.<br>See section 4.2.7 for additional requirements.<br>Bit definitions within this field are: Bit 0 2.5 GT/s Bit 1 5.0 GT/s Bit 2 8.0 GT/s Bits 6:3 RsvdP Bits in this field are RW if the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is Set, otherwise they are permitted to be hardwired to 0.<br>Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not Set.<br>The default value of this field is 000 0000b. |
| 8:2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1 | 0x0<br>RW | **Link Equalization Request Interrupt Enable (LNKEQREQINTEN):** When Set, this bit enables the generation of interrupt to indicate that the Link Equalization Request bit has been set.<br>This bit is not applicable and is reserved for Endpoints, PCI Express to PCI/PCIX bridges, and Upstream Ports of Switches. |
| 0 | 0x0<br>RW/V | **Perform Equalization (PE):** When this register is 1b and a 1b is written to the 'Link Retrain' register with 'Target Link Speed' set to 8 GT/s, the Downstream Port must perform Transmitter Equalization.<br>Software must guarantee that no transactions are pending in the link while the link is in Recovery when this bit is set to 1b.<br>This bit is not applicable and is reserved for Endpoints, PCI Express to PCI/PCIX bridges, and Upstream Ports of Switches. |

## 48.2.109 LANEERRSTS—Offset 208h

The Equalization Control Register consist of control fields required for per lane equalization and the number of entries in this register are sized by Max Link Width

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 208h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7 | 0x0<br>RW/1C/V/P | **Lane 7 Error (L7ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 6 | 0x0<br>RW/1C/V/P | **Lane 6 Error (L6ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 5 | 0x0<br>RW/1C/V/P | **Lane 5 Error (L5ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 4 | 0x0<br>RW/1C/V/P | **Lane 4 Error (L4ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 3 | 0x0<br>RW/1C/V/P | **Lane 3 Error (L3ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 2 | 0x0<br>RW/1C/V/P | **Lane 2 Error (L2ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 1 | 0x0<br>RW/1C/V/P | **Lane 1 Error (L1ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |
| 0 | 0x0<br>RW/1C/V/P | **Lane 0 Error (L0ERR):** A value of 1b indicates that a lane based error was detected on the corresponding PCI Express Lane Number. |

## 48.2.110 LANEEQCTL0—Offset 20Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 20Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB 001b: -7 dB<br>• 010b: -8 dB 011b: -9 dB<br>• 100b: -10 dB 101b: -11 dB<br>• 110b: -12 dB 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port.<br>The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of<br>• 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.111 LANEEQCTL1—Offset 20Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 20Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.112 LANEEQCTL2—Offset 210h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 210h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.113 LANEEQCTL3—Offset 212h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 212h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.114 LANEEQCTL4—Offset 214h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 214h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.115 LANEEQCTL5—Offset 216h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 216h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.116 LANEEQCTL6—Offset 218h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 218h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.117 LANEEQCTL7—Offset 21Ah

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 21Ah

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW/L | **Upstream Port Receiver Preset Hint (UPRPH):** Preset Hint for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 11:8 | 0x0<br>RW/L | **Upstream Port Transmitter Preset (UPTP):** Transmitter Preset for Upstream Port with the following encoding.<br>The Downstream Port must pass on this value in the EQ TS2's.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 dB for de-emphasis, 3.5 dB for preshoot<br>• 1001b: 0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>For a Upstream Port, this field reflects the latest Receiver Preset value requested from the Downstream Port for corresponding Lane.<br><br>**Power Well:** PRST |
| 7 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6:4 | 0x0<br>RW/L | **Downstream Port Receiver Preset Hint (DPRPH):** Receiver Preset Hint for Downstream Port. The Downstream Port may use this hint for receiver equalization.<br>• 000b: -6 dB<br>• 001b: -7 dB<br>• 010b: -8 dB<br>• 011b: -9 dB<br>• 100b: -10 dB<br>• 101b: -11 dB<br>• 110b: -12 dB<br>• 111b: Reserved<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |
| 3:0 | 0x0<br>RW/L | **Downstream Port Transmitter Preset (DPTP):** Transmitter Preset for an Downstream Port, with the following encoding.<br>• 0000b: -6 dB for de-emphasis, 0 dB for preshoot<br>• 0001b: -3.5 dB for de-emphasis, 0 dB for preshoot<br>• 0010b: -4.5 dB for de-emphasis, 0 dB for preshoot<br>• 0011b: -2.5 dB for de-emphasis, 0 dB for preshoot<br>• 0100b: 0 dB for de-emphasis, 0 dB for preshoot<br>• 0101b: 0 dB for de-emphasis, 2 dB for preshoot<br>• 0110b: 0 dB for de-emphasis, 2.5 dB for preshoot<br>• 0111b: -6 dB for de-emphasis, 3.5 dB for preshoot<br>• 1000b: -3.5 db for de-emphasis, 3.5 dB for preshoot<br>• 1001b:0 dB for de-emphasis, 3.5 dB for preshoot<br>• 1010b: The Transmitter Preset encoding of 1010b corresponds to the maximum Vtx-boost that the Transmitter advertises with FS and LF in both full swing and reduced swing modes with C-1 = 0 and C+1 = FS - (2*LF).<br>• All other encodings are reserved.<br>This field is not applicable and reserved for Endpoints, PCI Express to PCI/PCI-X bridges, and Upstream Ports of Switches.<br><br>**Power Well:** PRST |

## 48.2.118 XPPMDL0—Offset 280h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 280h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V | **PM data counter low value (CNTL):** Low order bits [31:0] for PM data counter[1:0]. |

## 48.2.119 XPPMDL1—Offset 284h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 284h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V | **PM data counter low value (CNTL):** Low order bits [31:0] for PM data counter[1:0]. |

## 48.2.120 XPPMCL0—Offset 288h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 288h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff RW/V | **PM compare low value (CMPL):** Low order bits [31:0] for PM compare register[1:0]. |

## 48.2.121 XPPMCL1—Offset 28Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 28Ch

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff RW/V | **PM compare low value (CMPL):** Low order bits [31:0] for PM compare register[1:0]. |

## 48.2.122 XPPMDH—Offset 290h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 290h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RW/V | **High Nibble PEX Counter1 value (CNTH1):** High order bits [35:32] of the 36-bit PM Data1 register. |
| 7:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3:0 | 0x0<br>RW/V | **High Nibble PEX Counter0 value (CNTH0):** High order bits [35:32] of the 36-bit PM Data0 register. |

## 48.2.123 XPPMCH—Offset 292h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 292h

**Default:** 0F0Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0xf<br>RW/V | **High Nibble PEX Compare1 value (CMPH1):** High order bits [35:32] of the 36-bit PM Compare1 register. |
| 7:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3:0 | 0xf<br>RW/V | **High Nibble PEX Compare0 value (CMPH0):** High order bits [35:32] of the 36-bit PM Compare0 register. |

## 48.2.124 XPPMR0—Offset 294h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 294h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Local/Global Event Select (LOCGLOBEVSEL):** This register bit allows for standalone cluster simulation.<br>Normally the global events are used as control signals for the performance monitors.<br>When this bit is set to1, the internal IPTRIG[0-3] signals are selected respectively in place of the external GE[0-3] signals.<br>***Note:*** It is possible to have one PERFMON use the internal events, while the other PERFMON uses the external events.<br>• 0:Global events are selected<br>• 1:Local events are selected |
| 30 | 0x0<br>RW | **Not greater than comparison (NOTGRTCMP):**<br>• 0: PMC will compare a greater than function. When clear the perfmon status will assert when the PMD is greater than the PMC.<br>• 1: PMC will compare with NOT (greater than) function. When set the perfmon status will assert when the PMD is less than or equal to the PMC.<br>CSPEC NOTE: This bit was not in Seaburg. |
| 29 | 0x0<br>RW | **Force PMD counter to add zero to input (FRCPMDADDZ):** This feature is used with the queue measurement bus.<br>When this bit is set the value on the queue measurement bus is added to zero so the result in PMD will always reflect the value from the queue measurement bus.<br>• 0: Do not add zero. Normal PerfMon operation.<br>• 1: Add zero with input queue bus. |
| 28 | 0x0<br>RW | **Latched Count Enable Select (LATCNTENSEL):**<br>• 0: Normal PM operation. Use CENS as count enable.<br>• 1: Use Latched count enable from queue empty events |
| 27 | 0x0<br>RW | **Reset Pulse Enable (RSTPULSEN):** Setting this bit will select a pulsed version of the reset signal source in the reset block.<br>• 0: Normal reset signaling<br>• 1: Select a pulsed reset from the reset signal sources. |
| 26:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:22 | 0x0<br>RW | **DFX Byte Lane Selection for PerfMon (DFXLNSEL):** This bit field is only active when LDES is asserted.<br>These bits select which set of 3 byte lanes from the 72 bit (9 byte lane) local debug bus is active for DFx event counting with the PerfMons.<br>• 00: Local Debug Byte Lanes 0-2<br>• 01: Local Debug Byte Lanes 3-5<br>• 10: Local Debug Byte Lanes 6-8<br>• 11: Reserved |
| 21 | 0x0<br>RW | **Local DFT Event Select (LDES):** This selection enables the local debug bus to be selected for event counting.<br>Enabling this selection changes the PMEVL,H register into 24bits of DFx masking and matching.<br>• 0: Disable DFx event monitoring<br>• 1: Enable DFx event monitoring. |
| 20:19 | 0x0<br>RW | **Event Group Selection (EGS):** Selects which event register to use for performance monitoring.<br>• 00: Bus events (XPPMEVL,H register)<br>• 01: Resource utilizations (in the XPPMER register)<br>• 10: Queue measurement (in the XPPMER register)<br>• 11: FIFO queue measurements (selects 1h on the queue measurement bus). |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18:17 | 0x0<br>RW | **Count Event Select (CNTEVSEL):** Selects the condition for incrementing the performance monitor counter.<br>• 00: Event source selected by PMEVL,H<br>• 01: Partner event status (max compare or overflow)<br>• 10: All clocks when enabled<br>• 11: Reserved |
| 16 | 0x0<br>RW | **Event Polarity Invert (EVPOLINV):** This bit inverts the polarity of the conditioned event signal.<br>• 0: No inversion<br>• 1: Invert the polarity of the conditioned event signal |
| 15:14 | 0x0<br>RW | **Count Mode (CNTMD):** This field sets how the events will be counted.<br>• 00: Count clocks for the duration of event signal is active.<br>• 01: Count rising edge events (for active low signals EVPOLINV must be set for correct measurements).<br>• 10: Count clocks continuously after the event was asserted. The latched state of this condition is cleared by PerfMon reset block, PMR.CNTRST bit, or XPPERFCON.GRST.<br>• 11: Enable queue threshold measuring.<br>This mode will reassign the event selection registers from both PerfMons, selects a modified count enable, and enables add/sub operation.<br>Selecting this mode will not allow normal PerfMon operations to occur.<br>To simplify the logic a strict assignment of functions is applied.<br>PM0's event selection registers are assigned to the add (increment) function and PM1's event selection registers are assigned to the subtract (decrement) function.<br>When both PM0 and PM1 events are asserted or when both are not asserted do not add or subtract values from the queue.<br>PM0Ev.....PM1Ev.....PM condition<br>....0.....0.....Do not add/sub<br>....1.....0.....Add queue value<br>....0.....1.....Sub queue value<br>....1.....1.....Do not add/sub<br><br>Setting this mode enables the PM event registers as described but it also enables the selection of a empty signal to be used a count enable signal.<br>The count enable is latched high continuously and only cleared with PM global reset or local counter reset.<br>The THRQMSRSTR bit field selects which bit[3] of each nibble in each byte lane that acts as a count enable.<br>Once this event occurs it is latched and queue counting begins.<br>Because this signal is latched a counter reset or PERCON global reset is required to start another measurement.<br>This latched condition is cleared by the PMR.CNTRST bit, or XPPERFCON.GRST. |
| 13:11 | 0x0<br>RW | **Counter enable source (CENS):** These bits identify which input enables the counter.<br>Default value disables counting.<br>• 000: Disabled<br>• 001: Local Count Enabled (LCEN). This bit is always a logic 1.<br>• 010: Partner counter's event status (max compare or overflow)<br>• 011: Header MatchOut signal<br>• 100: GE[0], from the Global Debug Event Block<br>• 101: GE[1], from the Global Debug Event Block<br>• 110: GE[2], from the Global Debug Event Block<br>• 111: GE[3], from the Global Debug Event Block<br>***Note:*** Address/Header MatchOut signal must align with PMEVL,H events for this to be effective.<br>***Note:*** The GEs used are dependent on bit 31 (LOCGLOBEVSEL). If bit 31 is used, local events will be used instead of global events. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Reset Event Select (RSTEVSEL):** This bit field will reset the PMD counter.<br>For details regarding usage and pair mode operation see Chapter 21.<br>• 000: No reset condition<br>• 001: Partner's event status: When the partner counter causes an event status condition to be activated, either by a counter overflow or max comparison, then this counter will reset and continue counting.<br>• 010: Partners PME register event: When the partner counter detects a match condition which meets its selected PME register qualifications, then this counter will reset and continue counting.<br>• 011: This PM counter's status output.<br>• 100: GE[0], from the Global Debug Event Block.<br>• 101: GE[1], from the Global Debug Event Block.<br>• 110: GE[2], from the Global Debug Event Block.<br>• 111: GE[3], from the Global Debug Event Block.<br>***Note:*** The GEs used are dependent on bit 31 (LOCGLOBEVSEL). If bit 31 is used, local events will be used instead of global events. |
| 7:6 | 0x0<br>RW | **Compare Mode (CMPMD):** This field defines how the PMC (compare) register is to be used.<br>• 00: compare mode disabled (PMC register not used) sig p = sig s<br>• 01: max compare only: The PMC register value is compared with the counter value. If the counter value is greater then the Compare Status (CMPSTAT) will be set. sig p = sig n<br>• 10: max compare with update of PMC at end of sample: The PMC register value is compared with the counter value, and if the counter value is greater, the PMC register is updated with the counter value. Note: The Compare Status field is not affected in this mode. sig p = sig n<br>• 11: Reserved |
| 5 | 0x0<br>RW | **PM Status Signal Output (PMSSIG):**<br>• 0: Level output from status/overflow signals.<br>• 1: Pulsed output from status/overflow signals. |
| 4:3 | 0x0<br>RW | **PerfMon Trigger Output (PTO):** This field selects what the signal is communicated to the chip's event logic structure.<br>• 00: No cluster trigger output from PerfMons or header match.<br>• 01: PM Status.<br>• 10: PM Event Detection.<br>• 11: Address Header Match |
| 2 | 0x0<br>RW/1C/V | **Compare Status (CMPSTAT):** This status bit captures a count compare event.<br>• 0: no event<br>• 1: count compare - PMD counter greater than PMC register when in compare mode.<br>This bit remains set once an event is reported even though the original condition is no longer valid. Writing a logic '1' clears the bit. |
| 1 | 0x0<br>RW/1C/V | **Overflow Status Bit (OVS):** This status bit captures the overflow event from the PMD counter.<br>This bit remains set once an event is reported even though the original condition is no longer valid. Writing a logic '1' clears the bit. |
| 0 | 0x0<br>RW | **Counter Reset (CNTRST):** Setting this bit resets the following bit fields.<br>This only applies to the set of registers for this PM block. Setting this bit resets the PMD counter, the associated adder storage register and the count mode state latch (see bits CNTMD) to the default state.<br>It does not change the state of this PMR register, the event selections, or the value in the compare register.<br>Note: This bit must be cleared by software, otherwise the counters remain in reset.<br>There is also a reset bit in the XPPERFCON register which clears all PM registers including the PMR. |

## 48.2.125 XPPMR1—Offset 298h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 298h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Local/Global Event Select (LOCGLOBEVSEL):** This register bit allows for standalone cluster simulation.<br>Normally the global events are used as control signals for the performance monitors.<br>When this bit is set to1, the internal IPTRIG[0-3] signals are selected respectively in place of the external GE[0-3] signals.<br>***Note:*** It is possible to have one PERFMON use the internal events, while the other PERFMON uses the external events.<br>• 0:Global events are selected<br>• 1:Local events are selected |
| 30 | 0x0<br>RW | **Not greater than comparison (NOTGRTCMP):**<br>• 0: PMC will compare a greater than function. When clear the perfmon status will assert when the PMD is greater than the PMC.<br>• 1: PMC will compare with NOT(greater than) function. When set the perfmon status will assert when the PMD is less than or equal to the PMC.<br>CSPEC NOTE: This bit was not in Seaburg. |
| 29 | 0x0<br>RW | **Force PMD counter to add zero to input (FRCPMDADDZ):** This feature is used with the queue measurement bus.<br>When this bit is set the value on the queue measurement bus is added to zero so the result in PMD will always reflect the value from the queue measurement bus.<br>• 0: Do not add zero. Normal PerfMon operation.<br>• 1: Add zero with input queue bus. |
| 28 | 0x0<br>RW | **Latched Count Enable Select (LATCNTENSEL):**<br>• 0: Normal PM operation. Use CENS as count enable.<br>• 1: Use Latched count enable from queue empty events |
| 27 | 0x0<br>RW | **Reset Pulse Enable (RSTPULSEN):** Setting this bit will select a pulsed version of the reset signal source in the reset block.<br>• 0: Normal reset signaling<br>• 1: Select a pulsed reset from the reset signal sources. |
| 26:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:22 | 0x0<br>RW | **DFX Byte Lane Selection for PerfMon (DFXLNSEL):** This bit field is only active when LDES is asserted.<br>These bits select which set of 3 byte lanes from the 72 bit (9 byte lane) local debug bus is active for DFx event counting with the PerfMons.<br>• 00: Local Debug Byte Lanes 0-2<br>• 01: Local Debug Byte Lanes 3-5<br>• 10: Local Debug Byte Lanes 6-8<br>• 11: Reserved |
| 21 | 0x0<br>RW | **Local DFT Event Select (LDES):** This selection enables the local debug bus to be selected for event counting.<br>Enabling this selection changes the PMEVL,H register into 24bits of DFx masking and matching.<br>• 0: Disable DFx event monitoring<br>• 1: Enable DFx event monitoring. |
| 20:19 | 0x0<br>RW | **Event Group Selection (EGS):** Selects which event register to use for performance monitoring.<br>• 00: Bus events (XPPMEVL,H register)<br>• 01: Resource utilizations (in the XPPMER register)<br>• 10: Queue measurement (in the XPPMER register)<br>• 11: FIFO queue measurements (selects 1h on the queue measurement bus). |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18:17 | 0x0<br>RW | **Count Event Select (CNTEVSEL):** Selects the condition for incrementing the performance monitor counter.<br>• 00: Event source selected by PMEVL,H<br>• 01: Partner event status (max compare or overflow)<br>• 10: All clocks when enabled<br>• 11: Reserved |
| 16 | 0x0<br>RW | **Event Polarity Invert (EVPOLINV):** This bit inverts the polarity of the conditioned event signal.<br>• 0: No inversion<br>• 1: Invert the polarity of the conditioned event signal |
| 15:14 | 0x0<br>RW | **Count Mode (CNTMD):** This field sets how the events will be counted.<br>• 00: Count clocks for the duration of event signal is active.<br>• 01: Count rising edge events (for active low signals EVPOLINV must be set for correct measurements).<br>• 10: Count clocks continuously after the event was asserted. The latched state of this condition is cleared by PerfMon reset block, PMR.CNTRST bit, or XPPERFCON.GRST.<br>• 11: Enable queue threshold measuring. This mode will reassign the event selection registers from both PerfMons, selects a modified count enable, and enables add/sub operation. Selecting this mode will not allow normal PerfMon operations to occur.<br>To simplify the logic a strict assignment of functions is applied.<br>PM0's event selection registers are assigned to the add (increment) function and PM1's event selection registers are assigned to the subtract (decrement) function.<br>When both PM0 and PM1 events are asserted or when both are not asserted do not add or subtract values from the queue.<br>PM0Ev.....PM1Ev.....PM condition<br>....0.....0.....Do not add/sub<br>....1.....0.....Add queue value<br>....0.....1.....Sub queue value<br>....1.....1.....Do not add/sub<br><br>Setting this mode enables the PM event registers as described but it also enables the selection of a empty signal to be used a count enable signal.<br>The count enable is latched high continuously and only cleared with PM global reset or local counter reset.<br>The THRQMSRSTR bit field selects which bit[3] of each nibble in each byte lane that acts as a count enable.<br>Once this event occurs it is latched and queue counting begins.<br>Because this signal is latched a counter reset or PERCON global reset is required to start another measurement.<br>This latched condition is cleared by the PMR.CNTRST bit, or XPPERFCON.GRST. |
| 13:11 | 0x0<br>RW | **Counter enable source (CENS):** These bits identify which input enables the counter.<br>Default value disables counting.<br>• 000: Disabled<br>• 001: Local Count Enabled (LCEN). This bit is always a logic 1.<br>• 010: Partner counter's event status (max compare or overflow)<br>• 011: Header MatchOut signal<br>• 100: GE[0], from the Global Debug Event Block<br>• 101: GE[1], from the Global Debug Event Block<br>• 110: GE[2], from the Global Debug Event Block<br>• 111: GE[3], from the Global Debug Event Block<br>***Note:*** Address/Header MatchOut signal must align with PMEVL,H events for this to be effective.<br>***Note:*** The GEs used are dependent on bit 31 (LOCGLOBEVSEL). If bit 31 is used, local events will be used instead of global events. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Reset Event Select (RSTEVSEL):** This bit field will reset the PMD counter.<br>For details regarding usage and pair mode operation see Chapter 21.<br>• 000: No reset condition<br>• 001: Partner's event status: When the partner counter causes an event status condition to be activated, either by a counter overflow or max comparison, then this counter will reset and continue counting.<br>• 010: Partners PME register event: When the partner counter detects a match condition which meets its selected PME register qualifications, then this counter will reset and continue counting.<br>• 011: This PM counter's status output.<br>• 100: GE[0], from the Global Debug Event Block.<br>• 101: GE[1], from the Global Debug Event Block.<br>• 110: GE[2], from the Global Debug Event Block.<br>• 111: GE[3], from the Global Debug Event Block.<br>***Note:*** The GEs used are dependent on bit 31 (LOCGLOBEVSEL). If bit 31 is used, local events will be used instead of global events. |
| 7:6 | 0x0<br>RW | **Compare Mode (CMPMD):** This field defines how the PMC (compare) register is to be used.<br>• 00: compare mode disabled (PMC register not used) sig p = sig s<br>• 01: max compare only: The PMC register value is compared with the counter value. If the counter value is greater then the Compare Status (CMPSTAT) will be set. sig p = sig n<br>• 10: max compare with update of PMC at end of sample: The PMC register value is compared with the counter value, and if the counter value is greater, the PMC register is updated with the counter value. Note, the Compare Status field is not affected in this mode. sig p = sig n<br>• 11: Reserved |
| 5 | 0x0<br>RW | **PM Status Signal Output (PMSSIG):**<br>• 0: Level output from status/overflow signals.<br>• 1: Pulsed output from status/overflow signals. |
| 4:3 | 0x0<br>RW | **PerfMon Trigger Output (PTO):** This field selects what the signal is communicated to the chip's event logic structure.<br>• 00: No cluster trigger output from PerfMons or header match.<br>• 01: PM Status.<br>• 10: PM Event Detection.<br>• 11: Address Header Match |
| 2 | 0x0<br>RW/1C/V | **Compare Status (CMPSTAT):** This status bit captures a count compare event.<br>• 0: no event<br>• 1: count compare - PMD counter greater than PMC register when in compare mode. This bit remains set once an event is reported even though the original condition is no longer valid. Writing a logic '1' clears the bit. |
| 1 | 0x0<br>RW/1C/V | **Overflow Status Bit (OVS):** This status bit captures the overflow event from the PMD counter.<br>This bit remains set once an event is reported even though the original condition is no longer valid.<br>Writing a logic '1' clears the bit. |
| 0 | 0x0<br>RW | **Counter Reset (CNTRST):** Setting this bit resets the following bit fields.<br>This only applies to the set of registers for this PM block.<br>Setting this bit resets the PMD counter, the associated adder storage register and the count mode state latch (see bits CNTMD) to the default state.<br>It does not change the state of this PMR register, the event selections, or the value in the compare register.<br>***Note:*** This bit must be cleared by software, otherwise the counters remain in reset. There is also a reset bit in the XPPERFCON register which clears all PM registers including the PMR. |

## 48.2.126 XPPMEVL0—Offset 29Ch

Selections in this register correspond to fields within the PEX packet header. Each field selection is ANDed with all other fields in this register including the XPPMEVH except where noted.

The qualifications for fields in this register are as follows: PMEV Match = ((IO_Cfg_Write_event + IO_Cfg_Read_event _+ Mem_Write_event + Mem_Read_event + Trusted_write_event + Trusted_read_event + General_event) and INOUTBND) + GESEL IO_Cfg_Write_event = (REQCMP[0] and CMPR[1] and RDWR[1] and DATALEN and (TTYP[2] + (TTYP[1] and CFGTYP))) IO_Cfg_Read_event = (REQCMP[0] and CMPR[1] and RDWR[0] and DATALEN and (TTYP[2] + (FMTTYP[1] and CFGTYP))) Mem_Write_event = (REQCMP[0] and CMPR[0] and RDWR[1] and DATALEN and TTYP[3] and LOCK and EXTADDR and SNATTR)

*Note:* An outbound memory write does not have a snoop attribute as an inbound memory write has. So the user should set SNATTR='11' for outbound memory write transaction event counting. Mem_Read_event = (REQCMP[0] and CMPR[1] and RDWR[0] and DATALEN and (( TTYP[3] and LOCK and EXTADDR and SNATTR) + TTYP[2] + (TTYP[1] and CFGTYP)))

*Note:* For outbound memory reads there is no concept of issuing a snoop cycle. The user should select SNATTR='11' for either snoop attribute. Msg_event = (TTYP[0] and DND) (INOUTBND[0] and (MatchEq) + (IOBND[1] and (MatchEq)

*Note:* Setting both bits in INOUTBND is acceptable however the performance data gathered will not be accurate since once one header can be counted at a time.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 29Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RW | **Data or No Data Attribute (DND):**<br>• x1: Request/completion/message packet with data<br>• 1x: Request/completion/message packet without data |
| 29:28 | 0x0<br>RW | **Snoop Attribute (SNATTR):**<br>• x1: Snoop required<br>• 1x: No snoop required<br>• 11: Either |
| 27:26 | 0x0<br>RW | **Request or Completion Packet Selection (REQCMP):**<br>• x1: Request packet<br>• 1x: Completion packet<br>• 11: Either |
| 25:24 | 0x0<br>RW | **Read or Write Selection (RDWR):**<br>• x1: Read<br>• 1x: Write<br>• 11: Either |
| 23:22 | 0x0<br>RW | **Completion Required (CMPR):** (Request packet only)<br>• x1: No completion required<br>• 1x: Completion required<br>• 11: Either |
| 21:20 | 0x0<br>RW | **Lock Attribute Selection (LOCK):**<br>• x1: No lock<br>• 1x: Lock<br>• 11: Either |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19:18 | 0x0<br>RW | **Extended Addressing Header (EXTADDR):**<br>• x1: 32b addressing<br>• 1x: 64b addressing<br>• 11: Either |
| 17:16 | 0x0<br>RW | **Configuration Type (CFGTYP):** (If Config cycle Space selected)<br>• x1: Type 0<br>• 1x: Type 1<br>• 11: Either |
| 15:11 | 0x0<br>RW | **Transaction Type Encoding (TTYP):** (ORed group)<br>• 1xxxx: Trusted<br>• x1xxx: Memory<br>• xx1xx: IO<br>• xxx1x: Configuration<br>• xxxx1: Messages<br>• 11111: Any transaction type |
| 10:4 | 0x0<br>RW | **Data Length (DATLEN):**<br>• 1xxxxxx: (129 to 256 bytes)<br>• x1xxxxx: (65 to 128 bytes)<br>• xx1xxxx: (33 to 64 bytes)<br>• xxx1xxx: (17 to 32 bytes)<br>• xxxx1xx: (9 to 16 bytes)<br>• xxxxx1x: (0 to 8 bytes)<br>• xxxxxx1: 0 bytes, used for a special zero length encoded packets<br>• 1111111: Any Data length |
| 3:0 | 0x0<br>RW | **Completion Status (CMPSTAT):** (for Completion Packet) or message encoding (for Request Packet).<br>• 1xxx: Completer abort<br>• x1xx: Configuration request retry status (only used for receive completions)<br>• xx1x: Unsupported request<br>• xxx1: Successful completion<br>• 1111: Any status |

## 48.2.127 XPPMEVL1—Offset 2A0h

Selections in this register correspond to fields within the PEX packet header. Each field selection is ANDed with all other fields in this register including the XPPMEVH except where noted. The qualifications for fields in this register are as follows: PMEV Match = ((IO_Cfg_Write_event + IO_Cfg_Read_event _+ Mem_Write_event + Mem_Read_event + Trusted_write_event + Trusted_read_event + General_event) and INOUTBND) + GESEL IO_Cfg_Write_event = (REQCMP[0] and CMPR[1] and RDWR[1] and DATALEN and (TTYP[2] + (TTYP[1] and CFGTYP))) IO_Cfg_Read_event = (REQCMP[0] and CMPR[1] and RDWR[0] and DATALEN and (TTYP[2] + (FMTTYP[1] and CFGTYP))) Mem_Write_event = (REQCMP[0] and CMPR[0] and RDWR[1] and DATALEN and TTYP[3] and LOCK and EXTADDR and SNATTR)

*Note:* An outbound memory write does not have a snoop attribute as an inbound memory write has. So the user should set SNATTR='11' for outbound memory write transaction event counting. Mem_Read_event = (REQCMP[0] and CMPR[1] and RDWR[0] and DATALEN and (( TTYP[3] and LOCK and EXTADDR and SNATTR) + TTYP[2] + (TTYP[1] and CFGTYP)))

*Note:* For outbound memory reads there is no concept of issuing a snoop cycle. The user should select SNATTR='11' for either snoop attribute. Msg_event = (TTYP[0] and DND) (INOUTBND[0] and (MatchEq) + (IOBND[1] and (MatchEq)

*Note:* Setting both bits in INOUTBND is acceptable however the performance data gathered will not be accurate since once one header can be counted at a time.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2A0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RW | **Data or No Data Attribute (DND):**<br>• x1: Request/completion/message packet with data<br>• 1x: Request/completion/message packet without data |
| 29:28 | 0x0<br>RW | **Snoop Attribute (SNATTR):**<br>• x1: Snoop required<br>• 1x: No snoop required<br>• 11: Either |
| 27:26 | 0x0<br>RW | **Request or Completion Packet Selection (REQCMP):**<br>• x1: Request packet<br>• 1x: Completion packet<br>• 11: Either |
| 25:24 | 0x0<br>RW | **Read or Write Selection (RDWR):**<br>• x1: Read<br>• 1x: Write<br>• 11: Either |
| 23:22 | 0x0<br>RW | **Completion Required (CMPR):** (Request packet only)<br>• x1: No completion required<br>• 1x: Completion required<br>• 11: Either |
| 21:20 | 0x0<br>RW | **Lock Attribute Selection (LOCK):**<br>• x1: No lock<br>• 1x: Lock<br>• 11: Either |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19:18 | 0x0<br>RW | **Extended Addressing Header (EXTADDR):**<br>• x1: 32b addressing<br>• 1x: 64b addressing<br>• 11: Either |
| 17:16 | 0x0<br>RW | **Configuration Type (CFGTYP):** (If Config cycle Space selected)<br>• x1: Type 0<br>• 1x: Type 1<br>• 11: Either |
| 15:11 | 0x0<br>RW | **Transaction Type Encoding (TTYP):** (ORed group)<br>• 1xxxx: Trusted<br>• x1xxx: Memory<br>• xx1xx: IO<br>• xxx1x: Configuration<br>• xxxx1: Messages<br>• 11111: Any transaction type |
| 10:4 | 0x0<br>RW | **Data Length (DATLEN):**<br>• 1xxxxxx: (129 to 256 bytes)<br>• x1xxxxx: (65 to 128 bytes)<br>• xx1xxxx: (33 to 64 bytes)<br>• xxx1xxx: (17 to 32 bytes)<br>• xxxx1xx: (9 to 16 bytes)<br>• xxxxx1x: (0 to 8 bytes)<br>• xxxxxx1: 0 bytes, used for a special zero length encoded packets<br>• 1111111: Any Data length |
| 3:0 | 0x0<br>RW | **Completion Status (CMPSTAT):** (for Completion Packet) or message encoding (for Request Packet).<br>• 1xxx: Completer abort<br>• x1xx: Configuration request retry status (only used for receive completions)<br>• xx1x: Unsupported request<br>• xxx1: Successful completion<br>• 1111: Any status |

## 48.2.128 XPPMEVH0—Offset 2A4h

Selections in this register correspond to fields within the PEX packet header. Each field selection is ANDed with all other fields in this register including the XPPMEVL except for the Global Event signals. These signals are OR'ed with any event in the XPPMEVL and enables for debug operations requiring the accumulation of specific debug signals.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2A4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:2 | 0x0<br>RW | **Global Event Selection (GESEL):** Selects which GE[3:0] is used for event counting.<br>This field is OR'd with other fields in this register.<br>The GEs cannot be qualified with other PerfMon signals.<br>If more than 1 GE is selected then the resultant event is the OR between each GE.<br>However, properly counting Global Event based on design, XP PM Response Control Register bit [13:11] CENS must be set to choose GE[3:0] and also bit[18:17] CNTEVSEL must be set to 2'b10.<br>• 1xxx: GE[3]<br>• x1xx: GE[2]<br>• xx1x: GE[1]<br>• xxx1: GE[0] |
| 1:0 | 0x0<br>RW | **Receive or Transmit Selection (RXTXSEL):** Selects which path to count transactions.<br>• 1x: Transmit<br>• x1: Receive (from PCI bus)<br>• 11: Either |

## 48.2.129 XPPMEVH1—Offset 2A8h

Selections in this register correspond to fields within the PEX packet header. Each field selection is ANDed with all other fields in this register including the XPPMEVL except for the Global Event signals. These signals are OR'ed with any event in the XPPMEVL and enables for debug operations requiring the accumulation of specific debug signals.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2A8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5:2 | 0x0<br>RW | **Global Event Selection (GESEL):** Selects which GE[3:0] is used for event counting.<br>This field is OR'd with other fields in this register.<br>The GEs cannot be qualified with other PerfMon signals.<br>If more than 1 GE is selected then the resultant event is the OR between each GE.<br>However, properly counting Global Event based on design, XP PM Response Control Register bit [13:11] CENS must be set to choose GE[3:0] and also bit[18:17] CNTEVSEL must be set to 2'b10.<br>• 1xxx: GE[3]<br>• x1xx: GE[2]<br>• xx1x: GE[1]<br>• xxx1: GE[0] |
| 1:0 | 0x0<br>RW | **Receive or Transmit Selection (RXTXSEL):** Selects which path to count transactions.<br>• 1x: Transmit<br>• x1: Receive (from PCI bus)<br>• 11: Either |

## 48.2.130 XPPMER0—Offset 2ACh

This register is used to select queuing structures for measurement. Use of this event register is mutually exclusive with the XPPMEVL,H registers. The Event Register Select field in the PMR register must select this register for to enable monitoring operations of the queues.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2ACh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Count Uncorrectable Errors (CNTUCERR):** Setting this bit enables the counting of uncorrectable errors by the perfmon counter.<br>Only uncorrectable errors that are not masked by the ERRUNCDETMSK register are counted.<br>Errors are counted on the ports enabled by the XPRSCA field in this register. |
| 23 | 0x0<br>RW | **Count Correctable Errors (CNTCERR):** Setting this bit enables the counting of correctable errors by the perfmon counter.<br>Only correctable errors that are not masked by the ERRCORDETMSK register are counted.<br>Errors are counted on the ports enabled by the XPRSCA field in this register. |
| 22 | 0x0<br>RW | **Tx L0s state utilization event (TXL0SU):**<br>• 0: No tx_l0s utilization selected<br>• 1: Enable tx_l0s utilization counting |
| 21 | 0x0<br>RW | **Rx L0s state utilization event (RXL0SU):**<br>• 0: No rx_l0s utilization selected<br>• 1: Enable rx_l0s utilization counting |
| 20:17 | 0x0<br>RW | **XP Resource Assignment (XPRSCA):** This selects which PCI-Express links are being monitored and against with XPPMEVL and XPPMEFH match registers and the queue measure select modes.<br>A logic 1 selects that PCIe link for monitoring.<br>• 1000: Select port 3 for monitoring.<br>• 0100: Select port 2 for monitoring.<br>• 0010: Select port 1 for monitoring.<br>• 0001: Select port 0 for monitoring.<br>If multiple bits are set, each port with a set bit will be counted (ie. you can count the total traffic on all ports). |
| 16:13 | 0x0<br>RW | **Link Utilization (LNKUTIL):** Link Utilization field counts clocks that a port is receiving data and checking for receiver errors.<br>It is used to calculate link reliability.<br>If one counter counts receiver errors and another counter counts the cycles the port was checking for receiver error, a bit error rate can be calculated.<br>• 0000: No event selected<br>• 1000: Port 3 of 4, where applicable (DSPs only)<br>• 0100: Port 2 of 4, where applicable (DSPs only)<br>• 0010: Port 1 of 4, where applicable (DSPs only)<br>• 0001: Port 0 of 4 (USP, DSPs, vSP/vRP)<br>If multiple bits are set, each port with a set bit will be counted (ie. you can count the total traffic on all ports)<br>For configs with less then 4 ports, the upper bits corresponding to non-existent ports have no effect |
| 12 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11:10 | 0x0<br>RW | **Queue Measure Select (QMSEL):**<br>• 00: Transaction Bandwidth Measurement. Counts data payload in dwords going in the direction programmed (RxTx, FC, VC, Port)<br>• 01: FIFO Queue Histogram Measurement. Uses the queue depth. (RxTx, FC, VC, Port, Hdr/Data)<br>• 10: Credit Consumed Histogram Measurement. Uses the credit consumed count. (RxTx, FC, VC, Port, Hdr/Data)<br>• 11: Reserved |
| 9 | 0x0<br>RW | **Header Data Select (HDSEL):**<br>• 0: Data Queue selected<br>• 1: Header Queue selected |
| 8 | 0x0<br>RW | **All VC Select (AVCSEL):**<br>• 0: Not all VC selected<br>• 1: All VC selected This bit will override VCSEL if selected |
| 7:5 | 0x0<br>RW | **VC Select (VCSEL):** This field selects which VC to count. Only one VC can be counted at a time. |
| 4:2 | 0x0<br>RW | **Flow Control Class Select (FCCSEL):** This field selects which flow class for resource event<br>• xx1: Posted<br>• x1x: Non-Posted<br>• 1xx: Completion |
| 1:0 | 0x0<br>RW | **Tx/Rx Select (TXRXSEL):** This field selects a Tx/Rx queue to monitor.<br>Either receiver or transmit direction for measurement.<br>• 1x: Transmit<br>• x1: Receive (from PCIe bus)<br>• 11: Either Transmit or Receive direction |

## 48.2.131 XPPMER1—Offset 2B0h

This register is used to select queuing structures for measurement. Use of this event register is mutually exclusive with the XPPMEVL,H registers. The Event Register Select field in the PMR register must select this register for to enable monitoring operations of the queues.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Count Uncorrectable Errors (CNTUCERR):** Setting this bit enables the counting of uncorrectable errors by the perfmon counter.<br>Only uncorrectable errors that are not masked by the ERRUNCDETMSK register are counted.<br>Errors are counted on the ports enabled by the XPRSCA field in this register. |
| 23 | 0x0<br>RW | **Count Correctable Errors (CNTCERR):** Setting this bit enables the counting of correctable errors by the perfmon counter.<br>Only correctable errors that are not masked by the ERRCORDETMSK register are counted.<br>Errors are counted on the ports enabled by the XPRSCA field in this register. |
| 22 | 0x0<br>RW | **Tx L0s state utilization event (TXL0SU):**<br>• 0: No tx_l0s utilization selected<br>• 1: Enable tx_l0s utilization counting |
| 21 | 0x0<br>RW | **Rx L0s state utilization event (RXL0SU):**<br>• 0: No rx_l0s utilization selected<br>• 1: Enable rx_l0s utilization counting |
| 20:17 | 0x0<br>RW | **XP Resource Assignment (XPRSCA):** This selects which PCI-Express links are being monitored and against with XPPMEVL and XPPMEFH match registers and the queue measure select modes.<br>A logic 1 selects that PCIe link for monitoring.<br>• 1000: Select port 3 for monitoring.<br>• 0100: Select port 2 for monitoring.<br>• 0010: Select port 1 for monitoring.<br>• 0001: Select port 0 for monitoring.<br>If multiple bits are set, each port with a set bit will be counted (ie. you can count the total traffic on all ports). |
| 16:13 | 0x0<br>RW | **Link Utilization (LNKUTIL):** Link Utilization field counts clocks that a port is receiving data and checking for receiver errors.<br>It is used to calculate link reliability.<br>If one counter counts receiver errors and another counter counts the cycles the port was checking for receiver error, a bit error rate can be calculated.<br>• 0000: No event selected<br>• 1000: Port 3 of 4, where applicable (DSPs only)<br>• 0100: Port 2 of 4, where applicable (DSPs only)<br>• 0010: Port 1 of 4, where applicable (DSPs only)<br>• 0001: Port 0 of 4 (USP, DSPs, vSP/vRP)<br>If multiple bits are set, each port with a set bit will be counted (ie. you can count the total traffic on all ports)<br>For configs with less then 4 ports, the upper bits corresponding to non-existent ports have no effect |
| 12 | 0x0<br>RO | **RSVD1:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11:10 | 0x0<br>RW | **Queue Measure Select (QMSEL):**<br>• 00: Transaction Bandwidth Measurement. Counts data payload in dwords going in the direction programmed (RxTx, FC, VC, Port)<br>• 01: FIFO Queue Histogram Measurement. Uses the queue depth. (RxTx, FC, VC, Port, Hdr/Data)<br>• 10: Credit Consumed Histogram Measurement. Uses the credit consumed count. (RxTx, FC, VC, Port, Hdr/Data)<br>• 11: Reserved |
| 9 | 0x0<br>RW | **Header Data Select (HDSEL):**<br>• 0: Data Queue selected<br>• 1: Header Queue selected |
| 8 | 0x0<br>RW | **All VC Select (AVCSEL):**<br>• 0: Not all VC selected<br>• 1: All VC selected This bit will override VCSEL if selected |
| 7:5 | 0x0<br>RW | **VC Select (VCSEL):** This field selects which VC to count. Only one VC can be counted at a time. |
| 4:2 | 0x0<br>RW | **Flow Control Class Select (FCCSEL):** This field selects which flow class for resource event<br>• xx1: Posted<br>• x1x: Non-Posted<br>• 1xx: Completion |
| 1:0 | 0x0<br>RW | **Tx/Rx Select (TXRXSEL):** This field selects a Tx/Rx queue to monitor.<br>Either receiver or transmit direction for measurement.<br>• 1x: Transmit<br>• x1: Receive (from PCIe bus)<br>• 11: Either Transmit or Receive direction |

## 48.2.132 XPPMDFXMAT0—Offset 2B4h

This register contains bits to enable matching of any event signal on the local debug bus in this cluster. Only 3 byte lanes are possible for PerfMon counting by selecting bits in PMR.DFXLNSEL. This register is active when the PMR.LDES is set.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2B4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:26 | 0x0<br>RW | **Divider (DIV):** See Table 3-16, 'Boolean Combination of Debug Signals' on page 109 of Chapter 3, 'DFD Event Logic' of the Seaburg DFT and Debug (DFx) CSPEC |
| 25:24 | 0x0<br>RW | **OR Function Locator (ORFLOC):** See Table 3-16, 'Boolean Combination of Debug Signals' on page 109 of Chapter 3, 'DFD Event Logic' of the Seaburg DFT and Debug (DFx)<br><br>CSPEC ORFLOC.....DIV.....Resultant Boolean Equation<br>00.....00.....C+B+A<br>00.....01.....(C+B)+A<br>00.....10.....C(B+A)<br>00.....11.....CBA<br>01.....01.....C+B+A<br>01.....11.....CB+A<br>10.....10.....C+B+A<br>10.....11.....C+BA<br>11.....11.....C+B+A<br>***Note:*** Assume that PMR.DFXLNSEL=00<br>1. Let A = a match on debug lane 0<br>2. Let B= a match on debug lane 1.<br>3. Let C= a match on debug lane 2. |
| 23:16 | 0x0<br>RW | **DFx Match on Event Signal Lane N+2 (DFXMAT2):** Reserved. |
| 15:8 | 0x0<br>RW | **DFx Match on Event Signal Lane N+1 (DFXMAT1):** Reserved. |
| 7:0 | 0x0<br>RW | **DFx Match on Event Signal Lane N (DFXMAT0):** Reserved. |

## 48.2.133 XPPMDFXMAT1—Offset 2B8h

This register contains bits to enable matching of any event signal on the local debug bus in this cluster. Only 3 byte lanes are possible for PerfMon counting by selecting bits in PMR.DFXLNSEL. This register is active when the PMR.LDES is set.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2B8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:26 | 0x0<br>RW | **Divider (DIV):** See Table 3-16, 'Boolean Combination of Debug Signals' on page 109 of Chapter 3, 'DFD Event Logic' of the Seaburg DFT and Debug (DFx) CSPEC |
| 25:24 | 0x0<br>RW | **OR Function Locator (ORFLOC):** See Table 3-16, 'Boolean Combination of Debug Signals' on page 109 of Chapter 3, 'DFD Event Logic' of the Seaburg DFT and Debug (DFx)<br><br>CSPEC ORFLOC.....DIV.....Resultant Boolean Equation<br>00.....00.....C+B+A<br>00.....01.....(C+B)+A<br>00.....10.....C(B+A)<br>00.....11.....CBA<br>01.....01.....C+B+A<br>01.....11.....CB+A<br>10.....10.....C+B+A<br>10.....11.....C+BA<br>11.....11.....C+B+A<br>***Note:*** Assume that PMR.DFXLNSEL=00<br>1. Let A = a match on debug lane 0.<br>2. Let B= a match on debug lane 1.<br>3. Let C= a match on debug lane 2. |
| 23:16 | 0x0<br>RW | **DFx Match on Event Signal Lane N+2 (DFXMAT2):** Reserved. |
| 15:8 | 0x0<br>RW | **DFx Match on Event Signal Lane N+1 (DFXMAT1):** Reserved. |
| 7:0 | 0x0<br>RW | **DFx Match on Event Signal Lane N (DFXMAT0):** Reserved. |

## 48.2.134 XPPMDFXMSK0—Offset 2BCh

This register contains bits to enable masking of any event signal on the local debug bus in this cluster. Only 3 byte lanes are possible for PerfMon counting by selecting bits in PMR.DFXLNSEL. This register is active when the PMR.LDES is set.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2BCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RW | **DFx Mask on Event Signal Lane N+2 (DFXMSK2):** Reserved. |
| 15:8 | 0x0<br>RW | **DFx Mask on Event Signal Lane N+1 (DFXMSK1):** Reserved. |
| 7:0 | 0x0<br>RW | **DFx Mask on Event Signal Lane N (DFXMSK0):** Reserved. |

## 48.2.135 XPPMDFXMSK1—Offset 2C0h

This register contains bits to enable masking of any event signal on the local debug bus in this cluster. Only 3 byte lanes are possible for PerfMon counting by selecting bits in PMR.DFXLNSEL. This register is active when the PMR.LDES is set.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RW | **DFx Mask on Event Signal Lane N+2 (DFXMSK2):** Reserved. |
| 15:8 | 0x0<br>RW | **DFx Mask on Event Signal Lane N+1 (DFXMSK1):** Reserved. |
| 7:0 | 0x0<br>RW | **DFx Mask on Event Signal Lane N (DFXMSK0):** Reserved. |

## 48.2.136 XPPERFCON—Offset 2C4h

The XPPERFCON register is a local performance monitor control register used to clear all of the counters or to assign which XP (PCIe) device participates in the performance monitoring event. All perfmon blocks will needs this register. We will perform ORing this register and global PERFCON from full chip for backup usage and self contained purpose.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1 | 0x0<br>RW | **Global Count Enable (GCE):** This bit is AND'ed with the PMR.CENS mux enable selection.<br>• 0: Disable all counters.<br>• 1: Enable local selection of CENS. |
| 0 | 0x0<br>RW | **Global Reset (GRST):** Setting this bit to a logic 1 will reset the following bit fields in all PerfMon blocks throughout the component.<br><br>xxxPMDx.CNT<br>xxxPMR.OVS<br>xxxPMR.CMPSTAT<br>xxxPMCx.PMCx registers, and count mode state latch selected by xxxPMR.CNTMD = '01'.<br>Where xxx is the cluster prefix name.<br><br>***Note:*** Software must clear this bit after setting it, otherwise the PerfMons remain in a reset state. |

## 48.2.137 XPPERFSTAT—Offset 2C8h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2C8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1 | 0x0<br>RW/1C/V | **XP cluster PM1 Status (XPPM1STAT):** This bit indicates the status of PM0 from either the compare or overflow status.<br>This is bit is cleared by writing a '1'. |
| 0 | 0x0<br>RW/1C/V | **XP cluster PM0 Status (XPPM0STAT):** This bit indicates the status of PM0 from either the compare or overflow status.<br>This is bit is cleared by writing a '1'. |

## 48.2.138 XPCDTHROTTLEP—Offset 2D0h

This register controls the withholding of posted credits for throttling and survivability. For PCIe ports, the receive direction applies to PCIe credits and the transmit direction applies to IOSF credits. For the vSP/RP where both sides are IOSF credits, the receive credits apply to the IDF credits. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL.TLCL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RW/P/L | **Tx Posted Request Data Throttling (TXPRD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 22:16 | 0x0<br>RW/P/L | **Tx Posted Request Header Throttling (TXPRH):** Remove this many credits from the available pool of credits for this type of transaction. |
| 15:7 | 0x0<br>RW/P/L | **Rx Posted Request Data Throttling (RXPRD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 6:0 | 0x0<br>RW/P/L | **Rx Posted Request Header Throttling (RXPRH):** Remove this many credits from the available pool of credits for this type of transaction. |

## 48.2.139 XPCDTHROTTLEN—Offset 2D4h

This register controls the withholding of non-posted credits for throttling and survivability. For PCIe ports, the receive direction applies to PCIe credits and the transmit direction applies to IOSF credits. For the vSP/RP where both sides are IOSF credits, the receive credits apply to the IDF credits. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL.TLCL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0 RW/P/L | **Tx Non-Posted Request Data Throttling (TXNPRD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 22:16 | 0x0 RW/P/L | **Tx Non-Posted Request Header Throttling (TXNPRH):** Remove this many credits from the available pool of credits for this type of transaction. |
| 15:7 | 0x0 RW/P/L | **Rx Non-Posted Request Data Throttling (RXNPRD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 6:0 | 0x0 RW/P/L | **Rx Non-Posted Request Header Throttling (RXNPRH):** Remove this many credits from the available pool of credits for this type of transaction. |

## 48.2.140 XPCDTHROTTLEC—Offset 2D8h

This register controls the withholding of completion credits for throttling and survivability. For PCIe ports, the receive direction applies to PCIe credits and the transmit direction applies to IOSF credits. For the vSP/RP where both sides are IOSF credits, the receive credits apply to the IDF credits. Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL.TLCL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0 RW/P/L | **Tx Completion Data Throttling (TXCPLD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 22:16 | 0x0 RW/P/L | **Tx Completion Header Throttling (TXCPLH):** Remove this many credits from the available pool of credits for this type of transaction. |
| 15:7 | 0x0 RW/P/L | **Rx Completion Data Throttling (RXCPLD):** Remove this many credits from the available pool of credits for this type of transaction. |
| 6:0 | 0x0 RW/P/L | **Rx Completion Header Throttling (RXCPLH):** Remove this many credits from the available pool of credits for this type of transaction. |

## 48.2.141 IOSFDEVCLKGCTL—Offset 2E4h

It is included for DMI EP in RC as an extension of Rootport definition.

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2E4h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block controls the delay before going into IDLE.<br>16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.142 SBDEVCLKGCTL—Offset 2E6h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 2E6h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x1<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins.<br>16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.143 VDHDRCAP—Offset 300h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 300h

**Default:** 0001000Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW/L | **Next Capability Offset (PCIeNextPtr):** Contains the offset of the next item in the capabilities list.<br><br>**Power Well:** PRST |
| 19:16 | 0x1<br>RO | **Capability Version (PCIeCapVersion):** These bits indicate the version number of the PCI Express capability structure. |
| 15:0 | 0xb<br>RO | **Extended Capability ID (PCIeCapID):** Identifies the function as a Vendor Specific capability. |

## 48.2.144 IBSTHDRCAP—Offset 304h

This register stores the version number of the capability item and other base information contained in the capability structure.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 304h

**Default:** 2A800001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x2a8<br>RO | **Vendor Specific Extended Capability Length (VSECLength):** These bits indicate the length of the PCI Express extended capability structure. |
| 19:16 | 0x0<br>RO | **Revision Number (VSECIDRev):** These bits indicate the Revision number of the PCI Express capability structure. |
| 15:0 | 0x1<br>RO | **Vendor Specific Extended Capability ID (VSECID):** Reserved. |

## 48.2.145 IBSTENGLKSPDCAP—Offset 310h

REUT-ENG-LSCAPDevice: 7

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 310h

**Default:** 00000007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **RSVD0:** Reserved |
| 29 | 0x0<br>RO | **Unique Speed (UniqSpd):** Reserved. |
| 28 | 0x0<br>RO | **LSCTR is Passive (SpdCtlPassive):** Link speed is set by Link Speed Control (IBISTENGLKSPDCTL) register or it is passive where REUT-ENG-LSCTR (IBISTENGLKSPDCTL) does not control link speed. |
| 27:26 | 0x0<br>RO | **RSVD1:** Reserved |
| 25:24 | 0x0<br>RO/V | **Number of Test Controllers (NumTestContr):** Need to update with respect to port type. |
| 23:16 | 0x0<br>RO | **RSVD2:** Reserved |
| 15:8 | 0x0<br>RO | **Link Speed Capability1 (LinkSpeedCap1):** This field indicates the link speed.<br>Valid only if Unique Speed is 1b and # of Test Controllers is ) 0.<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s)<br>• 00000100b: Gen 3 Speed (8 GT/s)<br>• Others: Reserved. |
| 7:0 | 0x7<br>RO/V | **Link Speed Capability (LinkSpeedCap):** This field indicates the link speed capabilities of test controller 0 (if Unique Speed = 1b) or all Test Controllers (if Unique Speed = 0).<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s - Never used)<br>• 00000011b: Gen 1 and Gen 2 Speeds<br>• 00000100b: Gen 3 Speed Only (Unique Speed = 1)<br>• 00000111b: Gen 1, Gen2, and Gen3 Speed<br>• Others: Reserved. |

## 48.2.146 IBSTENGLKSPDCTL—Offset 314h

REUT-ENG-LSCTR

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 314h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/L | **Link 3 Speed Select (Link3SpeedSel):** This field controls the link speed of Link 3 if defined. This field overrides target link speed field defined in PCI Express spec and forces the link to operate at specified speed when programmed to a non-zero value.<br>• 00000000b: Force link speed control off<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s)<br>• 00000100b: Gen 3 Speed (8 GT/s)<br>• Others: Reserved. |
| 23:16 | 0x0<br>RW/L | **Link 2 Speed Select (Link2SpeedSel):** This field controls the link speed of Link 2 if defined. This field overrides target link speed field defined in PCI Express spec and forces the link to operate at specified speed when programmed to a non-zero value.<br>• 00000000b: Force link speed control off<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s)<br>• 00000100b: Gen 3 Speed (8 GT/s)<br>• Others: Reserved. |
| 15:8 | 0x0<br>RW/L | **Link 1 Speed Select (Link1SpeedSel):** This field controls the link speed of Link 1 if defined. This field overrides target link speed field defined in PCI Express spec and forces the link to operate at specified speed when programmed to a non-zero value.<br>• 00000000b: Force link speed control off<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s)<br>• 00000100b: Gen 3 Speed (8 GT/s)<br>• Others: Reserved. |
| 7:0 | 0x0<br>RW/L | **Link 0 Speed Select (Link0SpeedSel):** This field controls the link speed of Link 0. This field overrides target link speed field defined in PCI Express spec and forces the link to operate at specified speed when programmed to a non-zero value.<br>• 00000000b: Force link speed control off<br>• 00000001b: Gen 1 Speed (2.5 GT/s)<br>• 00000010b: Gen 2 Speed (5 GT/s)<br>• 00000100b: Gen 3 Speed (8 GT/s)<br>• Others: Reserved. |

## 48.2.147 IBSTENGLKTXRXCFG—Offset 31Ch

REUT-ENG-LTRCON

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 31Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:26 | 0x0<br>RO | **RSVD0:** Reserved |
| 25 | 0x0<br>RO | **Link Automatic Configuration (LinkAutoConfiguration):** Link automatic configuration is not supported by this IBIST engine. |
| 24 | 0x0<br>RW/P/L | **IBIST Write Disable (ReutDisable):** When set, this bit will prevent software writes to the IBIST engine's PCI Express Extended Capability Register Structure to prevent unwanted usage of the IBIST engine.<br>Once set, the registers are unwritable until next Powergood reset.<br>***Note:*** At a Minimum, REUT-ENG-LSCTR.LinkSpeedSel, REUT-ENG-LTRCON.LinkControl, REUT-PAT-TEC.StartTest, and REUT-PH-CTR.Initialization Mode are cleared and locked. |
| 23:15 | 0x0<br>RO | **RSVD1:** Reserved |
| 14:8 | 0x0<br>RO | **Link Control 1 (LinkControl1):** This field is only valid if Unique Speed is a 1b and Number tof test controllers is ) 0. |
| 7 | 0x0<br>RO | **RSVD2:** Reserved |
| 6:0 | 0x0<br>RW/P/L | **Link Control (LinkControl):** This field indicates the number of links under the control of the IBIST Engine.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware.<br>All the odd bits can not be set by SW. Need to become RO-V to reflect LWCTRL register for downstream or Root ports. |

## 48.2.148 IBSTERRCRCSTS0—Offset 320h

REUT-ERR-CED0-3

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 320h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/1C/V/P | **CRC Error Overflow (CrcECC2Ovrflow):** Reserved. |
| 14:0 | 0x0<br>RW/1C/V/P | **CRC Error Counter (CrcECC2ErrCnt):** Reserved. |

## 48.2.149 IBSTERRRCRVSTS0—Offset 324h

REUT-ERR-ECCRCV0-3 IBSTERRRCRVSTS[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 324h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **Recovery Overflow (LinkRecovOverflow):** This bit when 1b, indicates the recovery counter has overflowed. |
| 30:16 | 0x0 RW/1C/V/P | **Recovery Counter (RecovCnt):** This field indicates the number of times the link(s) LTSSM state machine has entered the recovery state. |
| 15 | 0x0 RW/1C/V/P | **Receiver Error Overflow (RxErrCntOvrflow):** This bit when 1b, indicates the receive error counter has overflowed. |
| 14:0 | 0x0 RW/1C/V/P | **Receiver Error Counter (RxErrCnt):** This field indicates the number of times the link(s) has detected an 8b/10b receiver error. |

## 48.2.150 IBSTERRCRCSTS1—Offset 328h

REUT-ERR-CED0-3

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 328h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RO | **RSVD0:** Reserved |
| 15 | 0x0 RW/1C/V/P | **CRC Error Overflow (CrcECC2Ovrflow):** Reserved. |
| 14:0 | 0x0 RW/1C/V/P | **CRC Error Counter (CrcECC2ErrCnt):** Reserved. |

## 48.2.151 IBSTERRRCRVSTS1—Offset 32Ch

REUT-ERR-ECCRCV0-3 IBSTERRRCRVSTS[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 32Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **Recovery Overflow (LinkRecovOverflow):** This bit when 1b, indicates the recovery counter has overflowed. |
| 30:16 | 0x0 RW/1C/V/P | **Recovery Counter (RecovCnt):** This field indicates the number of times the link(s) LTSSM state machine has entered the recovery state. |
| 15 | 0x0 RW/1C/V/P | **Receiver Error Overflow (RxErrCntOvrflow):** This bit when 1b, indicates the receive error counter has overflowed. |
| 14:0 | 0x0 RW/1C/V/P | **Receiver Error Counter (RxErrCnt):** This field indicates the number of times the link(s) has detected an 8b/10b receiver error. |

## 48.2.152 IBSTERRCRCSTS2—Offset 330h

REUT-ERR-CED0-3

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 330h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RO | **RSVD0:** Reserved |
| 15 | 0x0 RW/1C/V/P | **CRC Error Overflow (CrcECC2Ovrflow):** Reserved. |
| 14:0 | 0x0 RW/1C/V/P | **CRC Error Counter (CrcECC2ErrCnt):** Reserved. |

## 48.2.153 IBSTERRRCRVSTS2—Offset 334h

REUT-ERR-ECCRCV0-3 IBSTERRRCRVSTS[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 334h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **Recovery Overflow (LinkRecovOverflow):** This bit when 1b, indicates the recovery counter has overflowed. |
| 30:16 | 0x0<br>RW/1C/V/P | **Recovery Counter (RecovCnt):** This field indicates the number of times the link(s) LTSSM state machine has entered the recovery state. |
| 15 | 0x0<br>RW/1C/V/P | **Receiver Error Overflow (RxErrCntOvrflow):** This bit when 1b, indicates the receive error counter has overflowed. |
| 14:0 | 0x0<br>RW/1C/V/P | **Receiver Error Counter (RxErrCnt):** This field indicates the number of times the link(s) has detected an 8b/10b receiver error. |

## 48.2.154 IBSTERRCRCSTS3—Offset 338h

REUT-ERR-CED0-3

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 338h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/1C/V/P | **CRC Error Overflow (CrcECC2Ovrflow):** Reserved. |
| 14:0 | 0x0<br>RW/1C/V/P | **CRC Error Counter (CrcECC2ErrCnt):** Reserved. |

## 48.2.155 IBSTERRRCRVSTS3—Offset 33Ch

REUT-ERR-ECCRCV0-3 IBSTERRRCRVSTS[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 33Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **Recovery Overflow (LinkRecovOverflow):** This bit when 1b, indicates the recovery counter has overflowed. |
| 30:16 | 0x0 RW/1C/V/P | **Recovery Counter (RecovCnt):** This field indicates the number of times the link(s) LTSSM state machine has entered the recovery state. |
| 15 | 0x0 RW/1C/V/P | **Receiver Error Overflow (RxErrCntOvrflow):** This bit when 1b, indicates the receive error counter has overflowed. |
| 14:0 | 0x0 RW/1C/V/P | **Receiver Error Counter (RxErrCnt):** This field indicates the number of times the link(s) has detected an 8b/10b receiver error. |

## 48.2.156 IBSTPHYLCTL0—Offset 344h

REUT-PH-CTR0-3 IBSTPHYLCTL[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 344h

**Default:** 00400000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x1<br>RW/P | **Enable Scramble with LFSR (EnableScrambleLFSR):** When set to 1b, this bit enables scrambling by setting the disable scrambling bit in TS1 to 0b.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 21:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13 | 0x0<br>RW/P | **Disable Auto-Compliance (DisableAutocompliance):** This bit when set to 1b does not allow the LTSSM state machine to transition from polling to compliance because squelch is enabled.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 12 | 0x0<br>RW/P | **Initialization Abort Freeze (InitAbortFreeze):** This bit when set to 1b freezes the LTSSM state machine at whenever a timeout occurs that would have caused the LTSSM state machine to exit to detect state.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 11 | 0x0<br>RO | **RSVD2:** Reserved |
| 10:8 | 0x0<br>RW/P/L | **Initialization Mode (InitializationMode):** This field indicated the initialization mode of the link.<br>• 000b: Normal Initialization<br>• 001b: Initialization towards Loopback state<br>• 010b: Tester Mode<br>• 011b: Initialization towards Loopback.Compliance Master<br>• 100b: Forced Entry to Compliance Master<br>• 101b: Forced Entry to Compliance Slave<br>• 110b: Initialization towards Compliance Receive<br>• 111b: Reserved.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. Note: Based on LKCON registers |
| 7:3 | 0x0<br>RO | **RSVD3:** Reserved |
| 2 | 0x0<br>RO | **Bypass Calibration (BypassCalib):**<br>***Note:*** This can change based on IO design. |
| 1 | 0x0<br>RO | **RSVD4:** Reserved |
| 0 | 0x0<br>RW/P | **Physical Layer Reset (PhyLayerReset):** This bit controls the reset assertion to the physical layer. Software is required to both set and clear this bit.<br>***Note:*** This bit is equivalent to the link disable bit in the link control register.<br>PCIe logic should 'OR' this bit and the link disable bit. |

## 48.2.157 IBSTPHYLCTL1—Offset 348h

REUT-PH-CTR0-3 IBSTPHYLCTL[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 348h

**Default:** 00400000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x1<br>RW/P | **Enable Scramble with LFSR (EnableScrambleLFSR):** When set to 1b, this bit enables scrambling by setting the disable scrambling bit in TS1 to 0b.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 21:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13 | 0x0<br>RW/P | **Disable Auto-Compliance (DisableAutocompliance):** This bit when set to 1b does not allow the LTSSM state machine to transition from polling to compliance because squelch is enabled.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 12 | 0x0<br>RW/P | **Initialization Abort Freeze (InitAbortFreeze):** This bit when set to 1b freezes the LTSSM state machine at whenever a timeout occurs that would have caused the LTSSM state machine to exit to detect state.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 11 | 0x0<br>RO | **RSVD2:** Reserved |
| 10:8 | 0x0<br>RW/P/L | **Initialization Mode (InitializationMode):** This field indicated the initialization mode of the link.<br>• 000b: Normal Initialization<br>• 001b: Initialization towards Loopback state<br>• 010b: Tester Mode<br>• 011b: Initialization towards Loopback.Compliance Master<br>• 100b: Forced Entry to Compliance Master<br>• 101b: Forced Entry to Compliance Slave<br>• 110b: Initialization towards Compliance Receive<br>• 111b: Reserved.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware.<br>***Note:*** Based on LKCON registers |
| 7:3 | 0x0<br>RO | **RSVD3:** Reserved |
| 2 | 0x0<br>RO | **Bypass Calibration (BypassCalib):**<br>***Note:*** This can change based on IO design. |
| 1 | 0x0<br>RO | **RSVD4:** Reserved |
| 0 | 0x0<br>RW/P | **Physical Layer Reset (PhyLayerReset):** This bit controls the reset assertion to the physical layer. Software is required to both set and clear this bit.<br>***Note:*** This bit is equivalent to the link disable bit in the link control register.<br>PCIe logic should 'OR' this bit and the link disable bit. |

## 48.2.158 IBSTPHYLCTL2—Offset 34Ch

REUT-PH-CTR0-3 IBSTPHYLCTL[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 34Ch

**Default:** 00400000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x1<br>RW/P | **Enable Scramble with LFSR (EnableScrambleLFSR):** When set to 1b, this bit enables scrambling by setting the disable scrambling bit in TS1 to 0b.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 21:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13 | 0x0<br>RW/P | **Disable Auto-Compliance (DisableAutocompliance):** This bit when set to 1b does not allow the LTSSM state machine to transition from polling to compliance because squelch is enabled.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 12 | 0x0<br>RW/P | **Initialization Abort Freeze (InitAbortFreeze):** This bit when set to 1b freezes the LTSSM state machine at whenever a timeout occurs that would have caused the LTSSM state machine to exit to detect state.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 11 | 0x0<br>RO | **RSVD2:** Reserved |
| 10:8 | 0x0<br>RW/P/L | **Initialization Mode (InitializationMode):** This field indicated the initialization mode of the link.<br>• 000b: Normal Initialization<br>• 001b: Initialization towards Loopback state<br>• 010b: Tester Mode<br>• 011b: Initialization towards Loopback.Compliance Master<br>• 100b: Forced Entry to Compliance Master<br>• 101b: Forced Entry to Compliance Slave<br>• 110b: Initialization towards Compliance Receive<br>• 111b: Reserved.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware.<br>***Note:*** Based on LKCON registers |
| 7:3 | 0x0<br>RO | **RSVD3:** Reserved |
| 2 | 0x0<br>RO | **Bypass Calibration (BypassCalib):**<br>***Note:*** This can change based on IO design. |
| 1 | 0x0<br>RO | **RSVD4:** Reserved |
| 0 | 0x0<br>RW/P | **Physical Layer Reset (PhyLayerReset):** This bit controls the reset assertion to the physical layer. Software is required to both set and clear this bit.<br>***Note:*** This bit is equivalent to the link disable bit in the link control register.<br>PCIe logic should 'OR' this bit and the link disable bit. |

## 48.2.159 IBSTPHYLCTL3—Offset 350h

REUT-PH-CTR0-3 IBSTPHYLCTL[1-3] are valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 350h

**Default:** 00400000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x1<br>RW/P | **Enable Scramble with LFSR (EnableScrambleLFSR):** When set to 1b, this bit enables scrambling by setting the disable scrambling bit in TS1 to 0b.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 21:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13 | 0x0<br>RW/P | **Disable Auto-Compliance (DisableAutocompliance):** This bit when set to 1b does not allow the LTSSM state machine to transition from polling to compliance because squelch is enabled.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 12 | 0x0<br>RW/P | **Initialization Abort Freeze (InitAbortFreeze):** This bit when set to 1b freezes the LTSSM state machine at whenever a timeout occurs that would have caused the LTSSM state machine to exit to detect state.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware. |
| 11 | 0x0<br>RO | **RSVD2:** Reserved |
| 10:8 | 0x0<br>RW/P/L | **Initialization Mode (InitializationMode):** This field indicated the initialization mode of the link.<br>• 000b: Normal Initialization<br>• 001b: Initialization towards Loopback state<br>• 010b: Tester Mode<br>• 011b: Initialization towards Loopback.Compliance Master<br>• 100b: Forced Entry to Compliance Master<br>• 101b: Forced Entry to Compliance Slave<br>• 110b: Initialization towards Compliance Receive<br>• 111b: Reserved.<br>This control bit or field requires a link re-initialization though the LTSSM Detect state before any action is taken by hardware.<br>***Note:*** Based on LKCON registers |
| 7:3 | 0x0<br>RO | **RSVD3:** Reserved |
| 2 | 0x0<br>RO | **Bypass Calibration (BypassCalib):**<br>***Note:*** This can change based on IO design. |
| 1 | 0x0<br>RO | **RSVD4:** Reserved |
| 0 | 0x0<br>RW/P | **Physical Layer Reset (PhyLayerReset):** This bit controls the reset assertion to the physical layer. Software is required to both set and clear this bit.<br>***Note:*** This bit is equivalent to the link disable bit in the link control register.<br>PCIe logic should 'OR' this bit and the link disable bit. |

## 48.2.160 IBSTTXLCTL—Offset 354h

REUT-PH-TDC

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 354h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/P | **Tx Data Lane Disable (TxDataLaneDisable):** This register field is used to selectively disable Tx lanes of the link.<br>• Bit 0: Lane 0 disable<br>• ...<br>• Bit 15: lane 15 disable.<br>It gates data from the REUT engine on to the lane. Disabled Lanes is not actually dead. |

## 48.2.161 IBSTTDSTS—Offset 358h

REUT-PH-TDS

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 358h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RO/V | **Tx Data Lane Detection Status (TxLaneDetectStat):** This register field indicates that the Tx lanes did not detect a receiver termination upon existing Detect LTSSM state.<br>• Bit 0: Lane 0 disable<br>• ...<br>• Bit 15: lane 15 disable. |

## 48.2.162 IBSTPLLNSTS0—Offset 36Ch

REUT-PH-LSR0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 36Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO/V | **Lane 7 Present State (Lane7PresState):** This field identifies the current state for lane 7 (or lane 15 for IBSTPLLNSTS1). |
| 27:24 | 0x0<br>RO/V | **Lane 6 Present State (Lane6PresState):** This field identifies the current state for lane 6 (or lane 14 for IBSTPLLNSTS1). |
| 23:20 | 0x0<br>RO/V | **Lane 5 Present State (Lane5PresState):** This field identifies the current state for lane 5 (or lane 13 for IBSTPLLNSTS1). |
| 19:16 | 0x0<br>RO/V | **Lane 4 Present State (Lane4PresState):** This field identifies the current state for lane 4 (or lane 12 for IBSTPLLNSTS1). |
| 15:12 | 0x0<br>RO/V | **Lane 3 Present State (Lane3PresState):** This field identifies the current state for lane 3 (or lane 11 for IBSTPLLNSTS1). |
| 11:8 | 0x0<br>RO/V | **Lane 2 Present State (Lane2PresState):** This field identifies the current state for lane 2 (or lane 10 for IBSTPLLNSTS1). |
| 7:4 | 0x0<br>RO/V | **Lane 1 Present State (Lane1PresState):** This field identifies the current state for lane 1 (or lane 9 for IBSTPLLNSTS1). |
| 3:0 | 0x0<br>RO/V | **Lane 0 Present State (Lane0PresState):** This field identifies the current state for lane 0 (or lane 8 for IBSTPLLNSTS1). |

## 48.2.163 IBSTPRECFG0—Offset 370h

REUT-PH-PRE0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 370h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf<br>RO/V | **Pre Configuration Lane 7 ID (PreConfigLane7ID):** This field identifies the Pre-configuration Link ID for lane 7 (or lane 15 for IBSTPRECFG1). |
| 27:24 | 0xf<br>RO/V | **Pre Configuration Lane 6 ID (PreConfigLane6ID):** This field identifies the Pre-configuration Link ID for lane 6 (or lane 14 for IBSTPRECFG1). |
| 23:20 | 0xf<br>RO/V | **Pre Configuration Lane 5 ID (PreConfigLane5ID):** This field identifies the Pre-configuration Link ID for lane 5 (or lane 13 for IBSTPRECFG1). |
| 19:16 | 0xf<br>RO/V | **Pre Configuration Lane 4 ID (PreConfigLane4ID):** This field identifies the Pre-configuration Link ID for lane 4 (or lane 12 for IBSTPRECFG1). |
| 15:12 | 0xf<br>RO/V | **Pre Configuration Lane 3 ID (PreConfigLane3ID):** This field identifies the Pre-configuration Link ID for lane 3 (or lane 11 for IBSTPRECFG1). |
| 11:8 | 0xf<br>RO/V | **Pre Configuration Lane 2 ID (PreConfigLane2ID):** This field identifies the Pre-configuration Link ID for lane 2 (or lane 10 for IBSTPRECFG1). |
| 7:4 | 0xf<br>RO/V | **Pre Configuration Lane 1 ID (PreConfigLane1ID):** This field identifies the Pre-configuration Link ID for lane 1 (or lane 9for IBSTPRECFG1). |
| 3:0 | 0xf<br>RO/V | **Pre Configuration Lane 0 ID (PreConfigLane0ID):** This field identifies the Pre-configuration Link ID for lane 0 (or lane 8 for IBSTPRECFG1). |

## 48.2.164 IBSTPOSTCFG0—Offset 374h

REUT-PH-POST0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 374h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf RO/V | **Post Configuration Lane 7 ID (PostConfigLane7ID):** This field identifies the Post-configuration Link ID for lane 7 (or lane 15 for IBSTPOSTCFG1). |
| 27:24 | 0xf RO/V | **Post Configuration Lane 6 ID (PostConfigLane6ID):** This field identifies the Post-configuration Link ID for lane 6 (or lane 14 for IBSTPOSTCFG1). |
| 23:20 | 0xf RO/V | **Post Configuration Lane 5 ID (PostConfigLane5ID):** This field identifies the Post-configuration Link ID for lane 5 (or lane 13 for IBSTPOSTCFG1). |
| 19:16 | 0xf RO/V | **Post Configuration Lane 4 ID (PostConfigLane4ID):** This field identifies the Post-configuration Link ID for lane 4 (or lane 12 for IBSTPOSTCFG1). |
| 15:12 | 0xf RO/V | **Post Configuration Lane 3 ID (PostConfigLane3ID):** This field identifies the Post-configuration Link ID for lane 3 (or lane 11 for IBSTPOSTCFG1). |
| 11:8 | 0xf RO/V | **Post Configuration Lane 2 ID (PostConfigLane2ID):** This field identifies the Post-configuration Link ID for lane 2 (or lane 10 for IBSTPOSTCFG1). |
| 7:4 | 0xf RO/V | **Post Configuration Lane 1 ID (PostConfigLane1ID):** This field identifies the Post-configuration Link ID for lane 1 (or lane 9 for IBSTPOSTCFG1). |
| 3:0 | 0xf RO/V | **Post Configuration Lane 0 ID (PostConfigLane0ID):** This field identifies the Post-configuration Link ID for lane 0 (or lane 8 for IBSTPOSTCFG1). |

## 48.2.165 IBSTPLLNSTS1—Offset 378h

REUT-PH-LSR0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 378h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO/V | **Lane 7 Present State (Lane7PresState):** This field identifies the current state for lane 7 (or lane 15 for IBSTPLLNSTS1). |
| 27:24 | 0x0<br>RO/V | **Lane 6 Present State (Lane6PresState):** This field identifies the current state for lane 6 (or lane 14 for IBSTPLLNSTS1). |
| 23:20 | 0x0<br>RO/V | **Lane 5 Present State (Lane5PresState):** This field identifies the current state for lane 5 (or lane 13 for IBSTPLLNSTS1). |
| 19:16 | 0x0<br>RO/V | **Lane 4 Present State (Lane4PresState):** This field identifies the current state for lane 4 (or lane 12 for IBSTPLLNSTS1). |
| 15:12 | 0x0<br>RO/V | **Lane 3 Present State (Lane3PresState):** This field identifies the current state for lane 3 (or lane 11 for IBSTPLLNSTS1). |
| 11:8 | 0x0<br>RO/V | **Lane 2 Present State (Lane2PresState):** This field identifies the current state for lane 2 (or lane 10 for IBSTPLLNSTS1). |
| 7:4 | 0x0<br>RO/V | **Lane 1 Present State (Lane1PresState):** This field identifies the current state for lane 1 (or lane 9 for IBSTPLLNSTS1). |
| 3:0 | 0x0<br>RO/V | **Lane 0 Present State (Lane0PresState):** This field identifies the current state for lane 0 (or lane 8 for IBSTPLLNSTS1). |

## 48.2.166 IBSTPRECFG1—Offset 37Ch

REUT-PH-PRE0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 37Ch

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf RO/V | **Pre Configuration Lane 7 ID (PreConfigLane7ID):** This field identifies the Pre-configuration Link ID for lane 7 (or lane 15 for IBSTPRECFG1). |
| 27:24 | 0xf RO/V | **Pre Configuration Lane 6 ID (PreConfigLane6ID):** This field identifies the Pre-configuration Link ID for lane 6 (or lane 14 for IBSTPRECFG1). |
| 23:20 | 0xf RO/V | **Pre Configuration Lane 5 ID (PreConfigLane5ID):** This field identifies the Pre-configuration Link ID for lane 5 (or lane 13 for IBSTPRECFG1). |
| 19:16 | 0xf RO/V | **Pre Configuration Lane 4 ID (PreConfigLane4ID):** This field identifies the Pre-configuration Link ID for lane 4 (or lane 12 for IBSTPRECFG1). |
| 15:12 | 0xf RO/V | **Pre Configuration Lane 3 ID (PreConfigLane3ID):** This field identifies the Pre-configuration Link ID for lane 3 (or lane 11 for IBSTPRECFG1). |
| 11:8 | 0xf RO/V | **Pre Configuration Lane 2 ID (PreConfigLane2ID):** This field identifies the Pre-configuration Link ID for lane 2 (or lane 10 for IBSTPRECFG1). |
| 7:4 | 0xf RO/V | **Pre Configuration Lane 1 ID (PreConfigLane1ID):** This field identifies the Pre-configuration Link ID for lane 1 (or lane 9for IBSTPRECFG1). |
| 3:0 | 0xf RO/V | **Pre Configuration Lane 0 ID (PreConfigLane0ID):** This field identifies the Pre-configuration Link ID for lane 0 (or lane 8 for IBSTPRECFG1). |

## 48.2.167 IBSTPOSTCFG1—Offset 380h

REUT-PH-POST0-1

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 380h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf<br>RO/V | **Post Configuration Lane 7 ID (PostConfigLane7ID):** This field identifies the Post-configuration Link ID for lane 7 (or lane 15 for IBSTPOSTCFG1). |
| 27:24 | 0xf<br>RO/V | **Post Configuration Lane 6 ID (PostConfigLane6ID):** This field identifies the Post-configuration Link ID for lane 6 (or lane 14 for IBSTPOSTCFG1). |
| 23:20 | 0xf<br>RO/V | **Post Configuration Lane 5 ID (PostConfigLane5ID):** This field identifies the Post-configuration Link ID for lane 5 (or lane 13 for IBSTPOSTCFG1). |
| 19:16 | 0xf<br>RO/V | **Post Configuration Lane 4 ID (PostConfigLane4ID):** This field identifies the Post-configuration Link ID for lane 4 (or lane 12 for IBSTPOSTCFG1). |
| 15:12 | 0xf<br>RO/V | **Post Configuration Lane 3 ID (PostConfigLane3ID):** This field identifies the Post-configuration Link ID for lane 3 (or lane 11 for IBSTPOSTCFG1). |
| 11:8 | 0xf<br>RO/V | **Post Configuration Lane 2 ID (PostConfigLane2ID):** This field identifies the Post-configuration Link ID for lane 2 (or lane 10 for IBSTPOSTCFG1). |
| 7:4 | 0xf<br>RO/V | **Post Configuration Lane 1 ID (PostConfigLane1ID):** This field identifies the Post-configuration Link ID for lane 1 (or lane 9 for IBSTPOSTCFG1). |
| 3:0 | 0xf<br>RO/V | **Post Configuration Lane 0 ID (PostConfigLane0ID):** This field identifies the Post-configuration Link ID for lane 0 (or lane 8 for IBSTPOSTCFG1). |

## 48.2.168 TLPCIECLKGCTL—Offset 8E0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8E0h

**Default:** 001Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **DIDT Mode (DIDT):** This bit when set to 1b will ensure that all clock gating is staggered by the amount of time determined by the Stagger Timer field in this register.<br><br>**Power Well:** PRST |
| 14:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8:0 | 0x1f<br>RW | **Stagger Timer (ST):** This field indicates the number of clocks that the IP block must be wait before the clock can be disable process begins.<br><br>**Power Well:** PRST |

## 48.2.169 TLPCIECLKGCCTL—Offset 8E2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8E2h

**Default:** 0055h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:6 | 0x1<br>RW | **Port Grant Count 3 (PGC3):** Reserved.<br><br>**Power Well:** PRST |
| 5:4 | 0x1<br>RW | **Port Grant Count 2 (PGC2):** Reserved.<br><br>**Power Well:** PRST |
| 3:2 | 0x1<br>RW | **Port Grant Count 1 (PGC1):** Reserved.<br><br>**Power Well:** PRST |
| 1:0 | 0x1<br>RW | **Port Grant Count 0 (PGC0):** Reserved.<br><br>**Power Well:** PRST |

## 48.2.170 PT0TXNCLKGCTL—Offset 8E4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8E4h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.171 PT1TXNCLKGCTL—Offset 8E6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8E6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.172 PT2TXNCLKGCTL—Offset 8E8h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8E8h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.173 PT3TXNCLKGCTL—Offset 8EAh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8EAh

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.174 XPTDEF2—Offset 8F8h

Lock Key bit is located in the Personality Lock Key Control Register ('PLKCTL.TLCL').

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 8F8h

**Default:** 00001D00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **MSI Address Violation Status (MSIAVS):** MSI address violation check status that does not target 0xFEEx_xxxx while MSIAVCD is not being set. This is only valid for RPs usage. |
| 30:19 | 0x0 RW/P/L | **Reserved ECO (RECO):** This field is reserved for ECO purpose. |
| 18 | 0x0 RW/P/L | **Hold Retrain during Surprise Removal Disable (HRSRD):** For Surprise Removal, EIP RPs will not allow retrain to occur until the SLOTSTS.PDCS bit has been cleared by software for the removal event. This bit is to disable the above behavior and allow the retrain to occur. This field is only applicable to RPs and should be treated as reserved for other devices. |
| 17 | 0x0 RW/P/L | **ECRC Malformed Error Precedence Disable (ECRCMFEPD):** This bit changes the error precedence if a TLP has both a ECRC error and is malform.<br>• When this bit is set to 1, the packet will be logged as an ECRC error.<br>• When set to 0, the packet will be logged as a malform TLP, which is compliant to the PCIe spec errata.<br>Only the type of error logged is affected. If the TLP is malform, it will be dropped with no credit return. If there is only an ECRC error detected on a packet, it will be dropped and credits will be returned per PCI-SIG Errata. |
| 16 | 0x0 RW/P/L | **MSI Address Violation Check Disable (MSIAVCD):** Disable MSI address violation check that does not target 0xFEEx_xxxx. This is only valid for RPs usage. |
| 15 | 0x0 RW/P/L | **Force Downstream Relaxed Order (FORCERODN):** Forces all downstream completions to follow relaxed ordering rules. This allows downstream completions to pass downstream posted requests. Note that setting this bit breaks the producer consumer model when the data is in the address space below the port and the flag is in the address space outside the port, because the read of the flag will not push the data writes down the port. The belief is that usage model is not used. |
| 14 | 0x0 RW/P/L | **Downstream Relaxed Order Disable (DNRODIS):** Disables the ability of downstream relaxed ordered completions from passing posted requests in the same direction. |
| 13 | 0x0 RW/P/L | **Upstream Relaxed Order Disable (UPRODIS):** Disables the ability of upstream relaxed ordered completions from passing posted requests in the same direction. |
| 12 | 0x1 RW/P/L | **IOSF tdparity check disable (TDPAR_CHK_DIS):** When set this bit disables the data parity check for incoming data into the IOSF Target block. The logic will ignore the data parity bit and will operate as though the parity bit did not exist. No errors will be logged and good DW based parity will be written into the data queue. |
| 11 | 0x1 RW/P/L | **IOSF tcparity check disable (TCPAR_CHK_DIS):** When set this bit disables the command parity check for incoming commands into the IOSF Target block. The logic will ignore the command parity bit and will operate as though the parity bit did not exist. No errors will be logged and good DW based parity will be written into the header queue. |
| 10 | 0x1 RW/P/L | **MCTP Message Bus Number Check Enable (MMBNCE):** When set, if the Requester ID bus number of an MCTP message request received on the PCIe link does not fall within the range between SCBN and SBBN, the MCTP message will be silently discarded. |
| 9 | 0x0 RW/P/L | **Completion credit advertisement enable (CPLCDT_EN):** Endpoints are required to advertise infinite completion credits. Setting this defeature bit breaks that rule and enables the advertisement of non-infinite completion credits and flow control updates for completion credits. This is only valid for project specific |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8 | 0x1<br>RW/P/L | **Enable Virtual Port IDs (EVPID):**<br>• 1: Enables SB Virtual Port ID mode with allocated VPIDs which is associated with ports for all the egress INTx and LTR messages based on project specific. We will assign (N-1) virtual sideband PortID's to associate with the physical PortID which is port 0. The physical one is the same one as being assigned by the router as we have now, and would be used for all non-INTx messages and INTx messages from port 0. However, the VPIDs would be used for INTx messages routing from port(N-1:1), respectively. This allows all INTx from N ports to be forwarded to the ILB accordingly, so ILB can differentiate the source of the INTx message.<br>• 0: Normal operation with single SB Port ID per cluster for INTx delivery. This is only valid for project specific with multi root ports. |
| 7 | 0x0<br>RW/P/L | **Enable Receive Completion Ordering (ERCO):** The default ordering is to allow received completions to pass received posted requests to avoid receiver overflow conditions if posted requests cannot be processed. Setting this bit will enable PCI Express ordering for completions, preventing completions from passing posted requests. This is only valid for project specific |
| 6 | 0x0<br>RW/P/L | **Enable MSI Generation Override (MSIGENOV):**<br>• 1: Enables Root Port MSI generation override even if PCICMD.BME = 0 for the root port device as required by PCI Express specification.<br>• 0: Normal operation for Root Port behavior. If PCICMD.BME = 0, the root port will not generate a MSI. This is only valid for project specific. |
| 5 | 0x0<br>RW/P/L | **Completion Combining Disable (CPLCOMBINE_D):** When a larger read request is split into smaller read requests, the completions for the small requests may be opportunistically combined into larger completions to enable higher bandwidth and link utilization, as long as packet formation rules are not violated. This bit disables the port from combining completions. This is only valid for project specific . |
| 4 | 0x0<br>RW/P/L | **Config Retry Disable (CRD):** Setting this bit disables the re-issuing of a configuration request if the request is completed with a configuration retry status (CRS). The completion with CRS will be returned to the requestor. |
| 3 | 0x0<br>RW/P/L | **Optional Unexpected Completion Check Disable (UCCHKD):** Setting this bit disables some optional unexpected checks on completions. Specifically, it disables the checking on the byte count and lower address fields. |
| 2 | 0x0<br>RW/P/L | **MCTP Disable Bus Check (MCTPDBC):** Setting this bit will disable the Bus 0 check for all MCTP VDM Type1 messages. The check is enabled by default as following:<br>• Upstream - ReqBus =/= 0<br>• Downstream - TgtBus =/= 0 |
| 1 | 0x0<br>RW/P/L | **MSI-X Single Message Vector (MSIXSMSGV):** This bit when set, causes only a single MSI-X vector to be generated if MSI-X is enabled. This bit affects the default value of the MSI-X Table Size field |
| 0 | 0x0<br>RW/P/L | **Global Stop Disable (GBLSTOPD):** When the design detects an uncorrectable internal error, a stop signal in the design asserts, blocking the processing of transactions in both directions in the transaction layer. Setting this bit, disables the stop mechanism, allowing the transactions to proceed as if no error occurred. |

## 48.2.175 PMIDLTMR—Offset 948h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 948h

**Default:** 00000306h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **ENPMTIMEOUT:** A number of pci-e cards have a particular erratum which causes them to send spurious PM requests.<br>When such spurious behavior happens, the northbridge starts the pm negotiation from its side and results in a system failure.<br>This bit when set enables an exit to recovery if the power management request did not terminate within 64-96us. |
| 30:28 | 0x0<br>RW/P | **PMDLLPCNT:** Only useful for upstream devices.<br>Do not take action based on seeing ASPM/PM DLLPs until 'PMDLLPCNT' DLLPs have been observed.<br>When set to 0, a single ASPM/PM DLLP will cause a response.<br>Used to work around broken devices that fail to flush all ASPM/PM DLLPs before entering the low power state and send spurious DLLPs upon return to L0 state. |
| 27:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x3<br>RW/P | **L0s_IDLE_TIMER:** Indicates the amount of idle time in [us], as defined in the PCI-E spec, after which a tx-L0s transition will be initiated.<br>Timers have an accuracy of -1/+0us. A setting of 0 will disable tx-L0s transitions. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4:0 | 0x6<br>RW/P | **L1_IDLE_TIMER:** Only used for Downstream devices.<br>Not useful for an RC. Downstream device will initiate ASPM L1 request after the specified number of us of idle time.<br>For RH can be used as a scratch register.<br>A setting of 0 can be used as a non-specd way of disabling ASPM L1. |

## 48.2.176 LLRBSIZE—Offset 98Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 98Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **RSVD0:** Reserved |
| 26:16 | 0x0<br>RO/V | **LUT_SIZE:** Number of physical entries in LUT (does not account for RTRYCTL.UNUSABLE_MAP_ENTRIES). Units are 'TLPs'. |
| 15:11 | 0x0<br>RO | **RSVD1:** Reserved |
| 10:0 | 0x0<br>RO/V | **LLRB_SIZE:** Number of symbol times of data in the LLRB (does not account for RTRYCTL.UNUSABLE_LLRB_ENTRIES). Units are symbol times. |

## 48.2.177 FRMERRMSK—Offset 99Ch

These errors in FRMERRMSK, STS only apply to Gen1/Gen2.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 99Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:10 | 0x0<br>RO | **RSVD0:** Reserved |
| 9 | 0x0<br>RW/P | **DUMP_IN_DLLP_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 8 | 0x0<br>RW/P | **DUMP_IN_TLP_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 7 | 0x0<br>RW/P | **ILL_STP_END_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 6 | 0x0<br>RW/P | **SDP_FRQ_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 5 | 0x0<br>RW/P | **STP_FRQ_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 4 | 0x0<br>RW/P | **GAP_SDP_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 3 | 0x0<br>RW/P | **GAP_STP_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 2 | 0x0<br>RW/P | **IDL_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 1 | 0x0<br>RW/P | **PAD_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |
| 0 | 0x0<br>RW/P | **DW1_ERR_MSK:** This bit masks reporting of the corresponding errors in the FRMERRSTS register. |

## 48.2.178 FRMERRSTS—Offset 99Eh

These errors in FRMERRMSK, STS only apply to Gen1/Gen2.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 99Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:10 | 0x0<br>RO | **RSVD0:** Reserved |
| 9 | 0x0<br>RW/1C/V/P | **DUMP_IN_DLLP_ERR:** Preprocessor block detected error in DLLP framing. |
| 8 | 0x0<br>RW/1C/V/P | **DUMP_IN_TLP_ERR:** Preprocessor block detected error in TLP framing. |
| 7 | 0x0<br>RW/1C/V/P | **ILL_STP_END_ERR:** STP/SDP/END/EDB seen in illegal lanes. |
| 6 | 0x0<br>RW/1C/V/P | **SDP_FRQ_ERR:** SDP seen too frequently in otherwise legal lanes. |
| 5 | 0x0<br>RW/1C/V/P | **STP_FRQ_ERR:** STP seen too frequently in otherwise legal lanes. |
| 4 | 0x0<br>RW/1C/V/P | **GAP_SDP_ERR:** Framing error: Gap seen between first packet and DLLP. |
| 3 | 0x0<br>RW/1C/V/P | **GAP_STP_ERR:** Framing error: Gap seen between first packet and TLP. |
| 2 | 0x0<br>RW/1C/V/P | **IDL_ERR:** Framing error: Logic idles not seen where expected. |
| 1 | 0x0<br>RW/1C/V/P | **PAD_ERR:** Framing error: PADs not seen where expected. |
| 0 | 0x0<br>RW/1C/V/P | **DW1_ERR:** Framing error in 1st DW of a TLP/DLLP. |

## 48.2.179 LLPCIECLKGCTL—Offset 9E0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9E0h

**Default:** 001Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **DIDT Mode (DIDT):** This bit when set to 1b will ensure that all clock gating is staggered by the amount of time determined by the Stagger Timer field in this register.<br><br>**Power Well:** PRST |
| 14:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8:0 | 0x1f<br>RW | **Stagger Timer (ST):** This field indicates the number of clocks that the IP block must be wait before the clock can be disable process begins.<br><br>**Power Well:** PRST |

## 48.2.180 LLPCIECLKGCCTL—Offset 9E2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9E2h

**Default:** 0055h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:6 | 0x1<br>RW | **Port Grant Count 3 (PGC3):** Reserved.<br><br>**Power Well:** PRST |
| 5:4 | 0x1<br>RW | **Port Grant Count 2 (PGC2):** Reserved.<br><br>**Power Well:** PRST |
| 3:2 | 0x1<br>RW | **Port Grant Count 1 (PGC1):** Reserved.<br><br>**Power Well:** PRST |
| 1:0 | 0x1<br>RW | **Port Grant Count 0 (PGC0):** Reserved.<br><br>**Power Well:** PRST |

## 48.2.181 PT0LNKCLKGCTL—Offset 9E4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9E4h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.182 PT1LNKCLKGCTL—Offset 9E6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9E6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.183 PT2LNKCLKGCTL—Offset 9E8h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9E8h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.184 PT3LNKCLKGCTL—Offset 9EAh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9EAh

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.185 CFGPCIECLKGCTL—Offset 9ECh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9ECh

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.186 CFGPCIECLKGCCTL—Offset 9EEh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + 9EEh

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1:0 | 0x1<br>RW | **Configuration Grant Count (CGC):** Note: The grant count needs to be greater than 0 for the arbitration to function. A grant count of 0 means no resources available and the requester will never be serviced.<br><br>**Power Well:** PRST |

## 48.2.187 LTSSMSMSTS—Offset A68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO/V | **LTSSMSTATEMAIN:** LTSSM State Main. |
| 27:24 | 0x0<br>RO/V | **LTSSMSTATESUB:** LTSSM State Sub State. |
| 23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22:20 | 0x0<br>RO/V | **LTSSMRXL0SSM:** LTSSM RxL0s SM |
| 19 | 0x0<br>RO | **RSVD1:** Reserved |
| 18:16 | 0x0<br>RO/V | **LTSSMRECEQSM:** LTSSM Recovery Equalization SM |
| 15:13 | 0x0<br>RO | **RSVD2:** Reserved |
| 12:5 | 0x0<br>RO/V | **LNKNUM:** Link Number |
| 4:3 | 0x0<br>RO/V | **RCVRATESUP:** Received Rate Support. |
| 2 | 0x0<br>RO/V | **LNKREVERSED:** Link Reversed |
| 1 | 0x0<br>RO/V | **DLACTIVE:** DL_Active |
| 0 | 0x0<br>RO/V | **LTSSMLNKUP:** LTSSM Link UP. |

## 48.2.188 LTSSMLNSTS0—Offset A70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO/V | **Rcvr SM Lane 7 (RCVRSMLN7):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Rcvr SM Lane 6 (RCVRSMLN6):** Reserved. |
| 23:20 | 0x0<br>RO/V | **Rcvr SM Lane 5 (RCVRSMLN5):** Reserved. |
| 19:16 | 0x0<br>RO/V | **Rcvr SM Lane 4 (RCVRSMLN4):** Reserved. |
| 15:12 | 0x0<br>RO/V | **Rcvr SM Lane 3 (RCVRSMLN3):** Reserved. |
| 11:8 | 0x0<br>RO/V | **Rcvr SM Lane 2 (RCVRSMLN2):** Reserved. |
| 7:4 | 0x0<br>RO/V | **Rcvr SM Lane 1 (RCVRSMLN1):** Reserved. |
| 3:0 | 0x0<br>RO/V | **Rcvr SM Lane 0 (RCVRSMLN0):** Reserved. |

## 48.2.189 LTSSMSTATELOG0—Offset A80h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A80h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **LAST8MAIN:**<br>• [3:0] present ltssm (or last if stopped) (newest state)<br>• [7:4] prior state<br>• [11:8] [15:12] [19:16] [23:20] [27:24] [31:28] oldest state logged |

## 48.2.190 LTSSMSTATELOG1—Offset A84h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **LAST8SUB:**<br>• [3:0] present ltssm (or last if stopped) (newest state)<br>• [7:4] prior state<br>• [11:8] [15:12] [19:16] [23:20] [27:24] [31:28] oldest state logged |

## 48.2.191 LTSSMSTATELOG2—Offset A88h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **LAST8RXL0S:**<br>• [3:0] present ltssm (or last if stopped) (newest state)<br>• [7:4] prior state<br>• [11:8] [15:12] [19:16] [23:20] [27:24] [31:28] oldest state logged |

## 48.2.192 LTSSMSTATELOGCTL—Offset A8Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A8Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8:1 | 0x0<br>RW/P | **STOP:** Match register to freeze logging [7:4] LTSSM sub state to match [3:0] LTSSM main state to match |
| 0 | 0x0<br>RW/P | **START:** 0--)1 transition clears log registers and restarts logging |

## 48.2.193 LTLNERRCTRL—Offset A94h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A94h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:4 | 0x0<br>RW/P | **SEL2:** select what state transition to log in LTLNERRLOG.STS2 |
| 3:0 | 0x0<br>RW/P | **SEL1:** select what state transition to log in LTLNERRLOG.STS1 |

## 48.2.194 LTLNERRSTS—Offset A96h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A96h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1 | 0x0<br>RW/1C/V/P | **CAPTURED2:** State transition seen and error results captured in LTLNERRLOG.STS2 |
| 0 | 0x0<br>RW/1C/V/P | **CAPTURED1:** State transition seen and error results captured in LTLNERRLOG.STS1 |

## 48.2.195 LTLNERRLOG—Offset A98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + A98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RO/V/P | **Lane 7 Error for STS2 (L7ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 22 | 0x0<br>RO/V/P | **Lane 6 Error for STS2 (L6ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 21 | 0x0<br>RO/V/P | **Lane 5 Error for STS2 (L5ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL12 |
| 20 | 0x0<br>RO/V/P | **Lane 4 Error for STS2 (L4ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 19 | 0x0<br>RO/V/P | **Lane 3 Error for STS2 (L3ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 18 | 0x0<br>RO/V/P | **Lane 2 Error for STS2 (L2ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 17 | 0x0<br>RO/V/P | **Lane 1 Error for STS2 (L1ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 16 | 0x0<br>RO/V/P | **Lane 0 Error for STS2 (L0ERRSTS2):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL2 |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RO/V/P | **Lane 7 Error for STS1 (L7ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 6 | 0x0<br>RO/V/P | **Lane 6 Error for STS1 (L6ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 5 | 0x0<br>RO/V/P | **Lane 5 Error for STS1 (L5ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 4 | 0x0<br>RO/V/P | **Lane 4 Error for STS1 (L4ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 3 | 0x0<br>RO/V/P | **Lane 3 Error for STS1 (L3ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 2 | 0x0<br>RO/V/P | **Lane 2 Error for STS1 (L2ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 1 | 0x0<br>RO/V/P | **Lane 1 Error for STS1 (L1ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |
| 0 | 0x0<br>RO/V/P | **Lane 0 Error for STS1 (L0ERRSTS1):** Per lane vector, 1 indicates bad lane on selected state transition chosen by LTLNERRCTRL.SEL1 |

## 48.2.196 UPCFGCSR—Offset AB8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AB8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x0<br>RO/V | **RCVDUPCFGCAP:** Received up configure support by link partner. |
| 21:16 | 0x0<br>RO/V | **ILW:** Initial Link Width (maximum width to up configure to).<br>ILW format is the same as MLW. |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/P | **UPCFGCAPEN:** This bit when set to 1 enables advertising dynamic width support in the training sequences.<br>When this bit is set 1 from 0, the link will move through Recovery to Config advertising upconfigure capability.<br>It will also ignore the settings in fields TLW and UPCFGEN at the time the bit is set. |
| 6:1 | 0x0<br>RW/P | **TLW:** This is the target link width used when mastering a dynamic width change.<br>If TLW is greater than the initial trained link width when LinkUp goes to 0--)1, then the initial link width is chosen as a cap. |
| 0 | 0x0<br>RW/V | **UPCFGEN:** When UPCFGCAPEN is set to 1 and this bit is set to 1, a dynamic width change is initiated if the TLW field is not equal to the current link width (if other end advertises support for it), this bit is cleared by hardware after making one attempt to change width.<br>On error, this bit will be cleared and TLW != NLW. If TLW==NLW and this bit is set, hardware will clear it without entering Recovery/Config. |

## 48.2.197 SOSCTL—Offset AC8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AC8h

**Default:** 00025A4Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22 | 0x0<br>RW/V/P | **SRISEN:** This field should be set prior to link training and may be loaded via fuse or strap.<br>When enabled, the high frequency SOS interval is defined in SRISSOSINT and the low frequency SOS interval is defined in SOSINTERVAL. |
| 21:11 | 0x4b<br>RW/P | **SRISSOSINT:** SOS interval used in SRIS mode (high frequency interval).<br>Same interval definition as the SOSINTERVAL field. |
| 10:0 | 0x24e<br>RW/P | **SOSINTERVAL:**<br>Symbol between SOS insertions For 2.5GTs/ and 5GT/s modes = val * 2 For 8GT/s mode = val * 8 |

## 48.2.198 G3SOSERRSTS—Offset ACCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + ACCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/1C/V/P | **LFSRERR:** 8GT/s SOS block LFSR state mis-compare. |
| 15:0 | 0x0<br>RW/1C/V/P | **PARERR:** 8GT/s SOS block parity error. |

## 48.2.199 PLPCIECLKGCTL—Offset AE0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AE0h

**Default:** 001Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **DIDT Mode (DIDT):** This bit when set to 1b will ensure that all clock gating is staggered by the amount of time determined by the Stagger Timer field in this register.<br><br>**Power Well:** PRST |
| 14:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8:0 | 0x1f<br>RW | **Stagger Timer (ST):** This field indicates the number of clocks that the IP block must be wait before the clock can be disable process begins.<br><br>**Power Well:** PRST |

## 48.2.200 PLPCIECLKGCCTL—Offset AE2h

The grant count needs to be greater than 0 for the arbitration to function. A grant count of 0 means no resources available and the requester will never be serviced.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AE2h

**Default:** 0055h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:6 | 0x1<br>RW | **Port Grant Count 3 (PGC3):** This field is valid only for multi-port configuration.<br><br>**Power Well:** PRST |
| 5:4 | 0x1<br>RW | **Port Grant Count 2 (PGC2):** This field is valid only for multi-port configuration.<br><br>**Power Well:** PRST |
| 3:2 | 0x1<br>RW | **Port Grant Count 1 (PGC1):** This field is valid only for multi-port configuration.<br><br>**Power Well:** PRST |
| 1:0 | 0x1<br>RW | **Port Grant Count 0 (PGC0):** Reserved.<br><br>**Power Well:** PRST |

## 48.2.201 PT0PHYCLKGCTL—Offset AE4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AE4h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.202 PT1PHYCLKGCTL—Offset AE6h

This register is valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AE6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.203 PT2PHYCLKGCTL—Offset AE8h

This register is valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AE8h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.204 PT3PHYCLKGCTL—Offset AEAh

This register is valid only for multi-port configuration.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + AEAh

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **Idle Clock Gating Enable (ICGE):** This bit when set enables clock gating to occur when the IP block is idle longer then the Idle Clock Timer.<br><br>**Power Well:** PRST |
| 14:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x10<br>RW | **Idle Clock Timer (ICT):** This field indicates the number of clocks that the IP block must be idle before the clock disable process begins. 16 (default value) is the minimum number of clocks recommended.<br><br>**Power Well:** PRST |

## 48.2.205 LTLGCAP—Offset B00h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B00h

**Default:** 36000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x3600<br>RO | **BAROFFSET:** Memory bar offset for ram entry 0 |
| 15:4 | 0x0<br>RO/V | **RAMSIZE:** Number of total ram entries. |
| 3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RO/V | **NUMSUBLOG:** Number of sub-loggers supported. |

## 48.2.206 LTLGCTRL—Offset B04h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B04h

**Default:** 00321000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **RSVD0:** Reserved |
| 22:20 | 0x3<br>RW/P | **SUBLOG3PT:** Port to assign to sub-logger 3. |
| 19 | 0x0<br>RO | **RSVD1:** Reserved |
| 18:16 | 0x2<br>RW/P | **SUBLOG2PT:** Port to assign to sub-logger 2. |
| 15 | 0x0<br>RO | **RSVD2:** Reserved |
| 14:12 | 0x1<br>RW/P | **SUBLOG1PT:** Port to assign to sub-logger 1. |
| 11 | 0x0<br>RO | **RSVD3:** Reserved |
| 10:8 | 0x0<br>RW/P | **SUBLOG0PT:** Port to assign to sub-logger 0. |
| 7:5 | 0x0<br>RO | **RSVD4:** Reserved |
| 4 | 0x0<br>RW/V/P | **CLEARALL:** Clear and restart all sub-loggers. |
| 3 | 0x0<br>RO | **RSVD5:** Reserved |
| 2:0 | 0x0<br>RW/P | **SPLITCTL:** Control of splitting logger if 1 sub log, ignored if 2 sub logs, only bit 0 used<br>[0] = 1, sublog0 owns full ram<br>[0] = 0, sublog0/1 split ram if 4 sub-logs<br>• 1-- = sublog0 owns full ram<br>• 011 = sublog0/2 split ram in half<br>• 001 = sublog0 gets half, sublog2/3 get a quarter each<br>• 010 = sublog0/1 get a quarter each, sublog2 gets half<br>• 000 = sublog0/1/2/3 get a quarter each |

## 48.2.207 LTLGSTART0—Offset B08h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B08h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RW/P | **STARTTRGMSK:**<br>If XSTOPTRGEN = 0, and any bit is set to 1, it will make that corresponding start trigger bit a don't care (always match).<br>If XSTOPTRGEN = 1, bits [19:18] are used and [31:20] reserved. Here is the definition for a and b which is the OR mask to dmona/b. a = bit[18] \| dmona b = bit[19] \| dmonb |
| 17:4 | 0x0<br>RW/P | **STARTTRG:**<br>If XSTOPTRGEN = 0, start trigger value, used in conjunction with the start trigger mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state.<br>If XSTOPTRGEN = 1, bits [6:4] are used, [17:7] reserved.<br>Here is the definition of the external trigger event using the mask and dmona/b values<br>• 000 - trigger (= (!a and !b)<br>• 001 - trigger (= ( a and !b)<br>• 010 - trigger (= (!a and b)<br>• 011 - trigger (= ( a and b)<br>• 100 - trigger (= (!a \| !b)<br>• 101 - trigger (= ( a \| !b)<br>• 110 - trigger (= (!a \| b)<br>• 111 - trigger (= ( a \| b) a = bit[18] \| dmona b = bit[19] \| dmonb |
| 3 | 0x0<br>RW/P | **STARTTRGEN:** If set, logging starts when the start trigger field and mask match the current LTSSM state/arc |
| 2 | 0x0<br>RW/1C/V/P | **STARTTRIGGERED:** If 1, the start trigger event was hit. |
| 1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/V/P | **START:** Enable logging. 0--)1 transition will clear log data If using the stop trigger, this bit will clear if the stop trigger event occurs. LTLGSTART[3:0] is based on one per sub-logger. |

## 48.2.208 LTLGSTART1—Offset B0Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B0Ch

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RW/P | **STARTTRGMSK:**<br>• If XSTOPTRGEN = 0, and any bit is set to 1, it will make that corresponding start trigger bit a don't care (always match).<br>• If XSTOPTRGEN = 1, bits [19:18] are used and [31:20] reserved.<br>Here is the definition for a and b which is the OR mask to dmona/b. a = bit[18] \| dmona b = bit[19] \| dmonb |
| 17:4 | 0x0<br>RW/P | **STARTTRG:**<br>If XSTOPTRGEN = 0, start trigger value, used in conjunction with the start trigger mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state<br>If XSTOPTRGEN = 1, bits [6:4] are used, [17:7] reserved.<br>Here is the definition of the external trigger event using the mask and dmona/b values<br>• 000 - trigger (= (!a and !b)<br>• 001 - trigger (= ( a and !b)<br>• 010 - trigger (= (!a and b)<br>• 011 - trigger (= ( a and b)<br>• 100 - trigger (= (!a \| !b)<br>• 101 - trigger (= ( a \| !b)<br>• 110 - trigger (= (!a \| b)<br>• 111 - trigger (= ( a \| b) a = bit[18] \| dmona b = bit[19] \| dmonb |
| 3 | 0x0<br>RW/P | **STARTTRGEN:** If set, logging starts when the start trigger field and mask match the current LTSSM state/arc |
| 2 | 0x0<br>RW/1C/V/P | **STARTTRIGGERED:** If 1, the start trigger event was hit. |
| 1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/V/P | **START:** Enable logging. 0--)1 transition will clear log data If using the stop trigger, this bit will clear if the stop trigger event occurs. LTLGSTART[3:0] is based on one per sub-logger. |

## 48.2.209 LTLGSTART2—Offset B10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RW/P | **STARTTRGMSK:**<br>• If XSTOPTRGEN = 0, and any bit is set to 1, it will make that corresponding start trigger bit a don't care (always match).<br>• If XSTOPTRGEN = 1, bits [19:18] are used and [31:20] reserved.<br>Here is the definition for a and b which is the OR mask to dmona/b. a = bit[18] \| dmona b = bit[19] \| dmonb |
| 17:4 | 0x0<br>RW/P | **STARTTRG:** If XSTOPTRGEN = 0, start trigger value, used in conjunction with the start trigger mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state If XSTOPTRGEN = 1, bits [6:4] are used, [17:7] reserved.<br>Here is the definition of the external trigger event using the mask and dmona/b values<br>• 000 - trigger (= (!a and !b)<br>• 001 - trigger (= ( a and !b)<br>• 010 - trigger (= (!a and b)<br>• 011 - trigger (= ( a and b)<br>• 100 - trigger (= (!a \| !b)<br>• 101 - trigger (= ( a \| !b)<br>• 110 - trigger (= (!a \| b)<br>• 111 - trigger (= ( a \| b) a = bit[18] \| dmona b = bit[19] \| dmonb |
| 3 | 0x0<br>RW/P | **STARTTRGEN:** If set, logging starts when the start trigger field and mask match the current LTSSM state/arc |
| 2 | 0x0<br>RW/1C/V/P | **STARTTRIGGERED:** If 1, the start trigger event was hit. |
| 1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/V/P | **START:** Enable logging. 0--)1 transition will clear log data If using the stop trigger, this bit will clear if the stop trigger event occurs. LTLGSTART[3:0] is based on one per sub-logger. |

## 48.2.210 LTLGSTART3—Offset B14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B14h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RW/P | **STARTTRGMSK:**<br>• If XSTOPTRGEN = 0, and any bit is set to 1, it will make that corresponding start trigger bit a don't care (always match).<br>• If XSTOPTRGEN = 1, bits [19:18] are used and [31:20] reserved.<br>Here is the definition for a and b which is the OR mask to dmona/b. a = bit[18] \| dmona b = bit[19] \| dmonb |
| 17:4 | 0x0<br>RW/P | **STARTTRG:**<br>If XSTOPTRGEN = 0, start trigger value, used in conjunction with the start trigger mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state If XSTOPTRGEN = 1, bits [6:4] are used, [17:7] reserved.<br>Here is the definition of the external trigger event using the mask and dmona/b values<br>• 000 - trigger (= (!a and !b)<br>• 001 - trigger (= ( a and !b)<br>• 010 - trigger (= (!a and b)<br>• 011 - trigger (= ( a and b)<br>• 100 - trigger (= (!a \| !b)<br>• 101 - trigger (= ( a \| !b)<br>• 110 - trigger (= (!a \| b)<br>• 111 - trigger (= ( a \| b) a = bit[18] \| dmona b = bit[19] \| dmonb |
| 3 | 0x0<br>RW/P | **STARTTRGEN:** If set, logging starts when the start trigger field and mask match the current LTSSM state/arc |
| 2 | 0x0<br>RW/1C/V/P | **STARTTRIGGERED:** If 1, the start trigger event was hit. |
| 1 | 0x0<br>RO | **RSVD0:** Reserved |
| 0 | 0x1<br>RW/V/P | **START:** Enable logging.<br>0--)1 transition will clear log data If using the stop trigger, this bit will clear if the stop trigger event occurs. LTLGSTART[3:0] is based on one per sub-logger. |

## 48.2.211 LTLGSTP0—Offset B18h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B18h

**Default:** 1E000009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30 | 0x0<br>RW/1C/V/P | **XTRIGGERED:** If 1, external stop trigger event occurred. |
| 29 | 0x0<br>RW/P | **XSTOPTRGEN:** Use external trigger event to stop logging, see LTLGCTL.STARTTRG and STARTTRGMSK for configuration details.<br>Enabling this does not disable the state match stop event and it does disable the state match start event (acts as if STARTTRGEN=0). |
| 28:15 | 0x3c00<br>RW/P | **STOPTRGMSK:** if any bit is set to 1, it will make that corresponding stopt trigger bit a don't care (always match) |
| 14:1 | 0x4<br>RW/P | **STOPTRG:** Stop trigger value, used in conjunction with the stop trigger mask [13:10]<br>State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x1<br>RW/P | **STOPTRGEN:** If set, logging stops when the stop trigger field and mask match the current LTSSM state/arc. LTLGSTP[3:0] is based on one per sub-logger. |

## 48.2.212 LTLGSTP1—Offset B1Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B1Ch

**Default:** 1E000009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30 | 0x0<br>RW/1C/V/P | **XTRIGGERED:** If 1, external stop trigger event occurred. |
| 29 | 0x0<br>RW/P | **XSTOPTRGEN:** Use external trigger event to stop logging, see LTLGCTL.STARTTRG and STARTTRGMSK for configuration details.<br>Enabling this does not disable the state match stop event and it does disable the state match start event (acts as if STARTTRGEN=0). |
| 28:15 | 0x3c00<br>RW/P | **STOPTRGMSK:** if any bit is set to 1, it will make that corresponding stop trigger bit a don't care (always match) |
| 14:1 | 0x4<br>RW/P | **STOPTRG:** Stop trigger value, used in conjunction with the stop trigger mask [13:10]<br>State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x1<br>RW/P | **STOPTRGEN:** If set, logging stops when the stop trigger field and mask match the current LTSSM state/arc. LTLGSTP[3:0] is based on one per sub-logger. |

## 48.2.213 LTLGSTP2—Offset B20h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B20h

**Default:** 1E000009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30 | 0x0 RW/1C/V/P | **XTRIGGERED:** If 1, external stop trigger event occurred. |
| 29 | 0x0 RW/P | **XSTOPTRGEN:** Use external trigger event to stop logging, see LTLGCTL.STARTTRG and STARTTRGMSK for configuration details.<br>Enabling this does not disable the state match stop event and it does disable the state match start event (acts as if STARTTRGEN=0). |
| 28:15 | 0x3c00 RW/P | **STOPTRGMSK:** if any bit is set to 1, it will make that corresponding stopt trigger bit a don't care (always match) |
| 14:1 | 0x4 RW/P | **STOPTRG:** Stop trigger value, used in conjunction with the stop trigger mask [13:10]<br>State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x1 RW/P | **STOPTRGEN:** If set, logging stops when the stop trigger field and mask match the current LTSSM state/arc. LTLGSTP[3:0] is based on one per sub-logger. |

## 48.2.214 LTLGSTP3—Offset B24h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B24h

**Default:** 1E000009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0 RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30 | 0x0 RW/1C/V/P | **XTRIGGERED:** If 1, external stop trigger event occurred. |
| 29 | 0x0 RW/P | **XSTOPTRGEN:** Use external trigger event to stop logging, see LTLGCTL.STARTTRG and STARTTRGMSK for configuration details.<br>Enabling this does not disable the state match stop event and it does disable the state match start event (acts as if STARTTRGEN=0). |
| 28:15 | 0x3c00 RW/P | **STOPTRGMSK:** if any bit is set to 1, it will make that corresponding stop trigger bit a don't care (always match) |
| 14:1 | 0x4 RW/P | **STOPTRG:** Stop trigger value, used in conjunction with the stop trigger mask [13:10]<br>State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x1 RW/P | **STOPTRGEN:** If set, logging stops when the stop trigger field and mask match the current LTSSM state/arc. LTLGSTP[3:0] is based on one per sub-logger. |

## 48.2.215 LTLGTMEV0—Offset B28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RW/P | **EVLVL:**<br>• 1: Event should be a level signal (high as long as compare condition is true)<br>• 0: Event should be a pulse |
| 22:6 | 0x0<br>RW/P | **COUNT:** The time value to compare with.<br>The value is in units defined by the UNIT field. |
| 5:3 | 0x0<br>RW/P | **CMP:**<br>• 00 = compare to equal the time value<br>• 01 = compare to greater than the time value<br>• 10 = compare to less than the time value<br>• 11 = reserved |
| 2 | 0x0<br>RW/P | **UNIT:**<br>• 1 = 256cc<br>• 0 = 1cc<br>Where cc is 1 pclk period and the user needs to be cognizant of designs that utilize variable (rate based) pclk. |
| 1 | 0x0<br>RW/P | **SMEGC:** Select which state matching register to use for the state matching portion.<br>The register must belong to the port set in the start register. |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable timer event using port defined in the start register.<br>LTLGMEV[3:0] is based on one per sub-logger. |

## 48.2.216 LTLGTMEV1—Offset B2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RW/P | **EVLVL:**<br>• 1: Event should be a level signal (high as long as compare condition is true)<br>• 0: Event should be a pulse |
| 22:6 | 0x0<br>RW/P | **COUNT:** The time value to compare with.<br>The value is in units defined by the UNIT field. |
| 5:3 | 0x0<br>RW/P | **CMP:**<br>• 00 = compare to equal the time value<br>• 01 = compare to greater than the time value<br>• 10 = compare to less than the time value<br>• 11 = reserved |
| 2 | 0x0<br>RW/P | **UNIT:**<br>• 1 = 256cc<br>• 0 = 1cc<br>Where cc is 1 pclk period and the user needs to be cognizant of designs that utilize variable (rate based) pclk. |
| 1 | 0x0<br>RW/P | **SMEGC:** Select which state matching register to use for the state matching portion.<br>The register must belong to the port set in the start register. |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable timer event using port defined in the start register.<br>LTLGMEV[3:0] is based on one per sub-logger. |

## 48.2.217 LTLGTMEV2—Offset B30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RW/P | **EVLVL:**<br>• 1: Event should be a level signal (high as long as compare condition is true)<br>• 0: Event should be a pulse |
| 22:6 | 0x0<br>RW/P | **COUNT:** The time value to compare with. The value is in units defined by the UNIT field. |
| 5:3 | 0x0<br>RW/P | **CMP:**<br>• 00 = compare to equal the time value<br>• 01 = compare to greater than the time value<br>• 10 = compare to less than the time value<br>• 11 = reserved |
| 2 | 0x0<br>RW/P | **UNIT:**<br>• 1 = 256cc<br>• 0 = 1cc<br>Where cc is 1 pclk period and the user needs to be cognizant of designs that utilize variable (rate based) pclk. |
| 1 | 0x0<br>RW/P | **SMEGC:** Select which state matching register to use for the state matching portion.<br>The register must belong to the port set in the start register. |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable timer event using port defined in the start register. LTLGMEV[3:0] is based on one per sub-logger. |

## 48.2.218 LTLGTMEV3—Offset B34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B34h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RW/P | **EVLVL:**<br>• 1: Event should be a level signal (high as long as compare condition is true)<br>• 0: Event should be a pulse |
| 22:6 | 0x0<br>RW/P | **COUNT:** The time value to compare with. The value is in units defined by the UNIT field. |
| 5:3 | 0x0<br>RW/P | **CMP:**<br>• 00 = compare to equal the time value<br>• 01 = compare to greater than the time value<br>• 10 = compare to less than the time value<br>• 11 = reserved |
| 2 | 0x0<br>RW/P | **UNIT:**<br>1 = 256cc<br>0 = 1cc<br>Where cc is 1 pclk period and the user needs to be cognizant of designs that utilize variable (rate based) pclk. |
| 1 | 0x0<br>RW/P | **SMEGC:** Select which state matching register to use for the state matching portion.<br>The register must belong to the port set in the start register. |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable timer event using port defined in the start register.<br>LTLGMEV[3:0] is based on one per sub-logger. |

## 48.2.219 LTLGRDCT—Offset B40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17:16 | 0x0<br>RW/P | **LOGSEL:** Select which sub-logger to read. |
| 15 | 0x0<br>RW/P | **AUTOINC:**<br>• 1: Every time a read to LTLGRES happens, the read index increment automatically<br>• 0: manual movement of the read index is required |
| 14:6 | 0x0<br>RO | **RSVD1:** Reserved |
| 5:0 | 0x0<br>RW/V/P | **RDINDEX:** The value in this field will point the LTLGRES register to a particular log entry. 0 is the latest entry 63 is oldest entry |

## 48.2.220 LTLGRES—Offset B44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B44h

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO/V/P | **ENTRY:** [31] Time granularity bit<br>• 1=512ns<br>• 0=2ns<br>Switch between modes is done when [30:14] increments past 1FFFFh [30:14] time unit count see bit 31 for unit definition [13:10]<br>State exit arc<br>• 0=state not exited<br>• 1..15=see state specific map [9:8] RxL0s State [7:4] LTSSM Main state [3:0] LTSSM Sub state |

## 48.2.221 LTLGSMEV0—Offset B48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:15 | 0x0<br>RW/P | **MASK:** if any bit is set to 1, it will make that corresponding trigger bit a don't care (always match) |
| 14:1 | 0x0<br>RW/P | **VALUE:** trigger value, used in conjunction with the mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable the trigger on match |

## 48.2.222 LTLGSMEV1—Offset B4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V/P | **TRIGGERED:** If 1, the trigger event was hit. |
| 30:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:15 | 0x0<br>RW/P | **MASK:** if any bit is set to 1, it will make that corresponding trigger bit a don't care (always match) |
| 14:1 | 0x0<br>RW/P | **VALUE:** trigger value, used in conjunction with the mask [13:10] State exit arc [9:8] RxL0s state [7:4] LTSSM main state [3:0] LTSSM sub state |
| 0 | 0x0<br>RW/P | **ENABLE:** Enable the trigger on match |

## 48.2.223 NFTSCOMCLK—Offset B50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B50h

**Default:** 00FFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0xff<br>RW/P | **GEN3NFTS:** Number of FTS blocks required for RxL0s in common clock mode |
| 15:8 | 0xff<br>RW/P | **GEN2NFTS:** Number of FTS ordered sets required for RxL0s in common clock mode |
| 7:0 | 0xff<br>RW/P | **GEN1NFTS:** Number of FTS ordered sets required for RxL0s in common clock mode |

## 48.2.224 NFTSDSCCLK—Offset B54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B54h

**Default:** 00FFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0xff<br>RW/P | **GEN3NFTS:** Number of FTS blocks required for RxL0s in non-common clock mode |
| 15:8 | 0xff<br>RW/P | **GEN2NFTS:** Number of FTS ordered sets required for RxL0s in non-common clock mode |
| 7:0 | 0xff<br>RW/P | **GEN1NFTS:** Number of FTS ordered sets required for RxL0s in non-common clock mode |

## 48.2.225 LEKBERR—Offset B58h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B58h

**Default:** 2000FFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **Enable Leaky Bucket Error Counter (G3LBEEN):** Enable Leaky Bucket Error Counters for gen3, this allows H/W to request re-equalization based on error rate. |
| 30 | 0x0<br>RO | **RSVD0:** Reserved |
| 29:25 | 0x10<br>RW/P | **Gen3 Leaky Bucket Error Threshold (G3ERRTHRESH):** Leaky Bucket error threshold used for 8GT/s and above.<br>When the error count for a lane passes this threshold, that lane will be flagged as faulty. |
| 24:20 | 0x0<br>RW/P | **Leaky Bucket Error Threshold (ERRTHRESH):** Sets the threshold for leaky bucket error handling.<br>When the error count for a lane passes this threshold, that lane will be flagged as faulty. |
| 19:0 | 0xffff<br>RW/P | **Aggregate error for leaky bucket error handling (AGGRERR):** The error count for the leaky bucket algorithm increments by one when AGGRERR number of errors have been encountered during the specified time period. |

## 48.2.226 EXPBERR0—Offset B5Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B5Ch

**Default:** FFFFFFFFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RW/P | **EXP_BER:** [31:0] Expected BER for leaky bucket error handling [31:0].<br>Selects the time window that is sampled for aggregate errors. |

## 48.2.227 EXPBERR1—Offset B60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B60h

**Default:** 02000007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x2<br>RW/P | **G3AGGRERR:** Aggregate error for gen3. The error count for the leaky bucket algorithm increments by one when AGGRERR number of errors have been encountered during the specified time period. |
| 23:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17:0 | 0x7<br>RW/P | **EXP_BER:** [49:32] Expected BER for leaky bucket error handling [49:32].<br>Selects the time window that is sampled for aggregate errors. |

## 48.2.228 LEKBPROERR—Offset B64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RW/1C/V/P | **G3ERRLNSTS:** Feature to check status of lane for Gen3.<br>Bit is set when the current error rate exceeds the set error threshold. |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7:0 | 0x0<br>RW/1C/V/P | **Lane Error Limit Status (ERRLNSTS):** Feature to check status of lane.<br>Bit is set when the current error rate exceeds the set error threshold. |

## 48.2.229 SPEEDCTL—Offset B68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B68h

**Default:** 02001821h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **RSVD0:** Reserved |
| 29 | 0x0<br>RW/V | **LBEG3DIS:** When set to 1, 8GT/s and higher rates are disabled.<br>Hardware or software may set this bit. Hardware will set it when the LBE is enabled to degrade link speed and the LBE threshold is tripped.<br>Software may set or clear at its will.<br>The register is sticky through in-band resets. |
| 28 | 0x0<br>RW/V | **LBEG2DIS:** When set to 1, 5GT/s and higher rates are disabled.<br>Hardware or software may set this bit.<br>Hardware will set it when the LBE is enabled to degrade link speed and the LBE threshold is tripped.<br>Software may set or clear at its will.<br>The register is sticky through in-band resets. |
| 27:26 | 0x0<br>RO | **RSVD1:** Reserved |
| 25 | 0x1<br>RW/P | **LBEG3DEGRADEEN:** Enable LBE to trigger a speed degrade from 8GT/s operation.<br>The 8GT/s mode and higher is disabled until LBEG3DIS is cleared. |
| 24 | 0x0<br>RW/P | **LBEG2DEGRADEEN:** Enable LBE to trigger a speed degrade from 5GT/s operation.<br>The 5GT/s mode and higher is disabled until LBEG2DIS is cleared. |
| 23:21 | 0x0<br>RO | **RSVD2:** Reserved |
| 20 | 0x0<br>RW/V | **RL:** Retrain Link. This bit has the same functionality as the LinkCtrl.Retrain does for downstream ports.<br>It is intended to be used when initiating a software based speed change. |
| 19:16 | 0x0<br>RO | **UPTLS:** unused in downstream ports |
| 15:13 | 0x0<br>RW/P | **SPDCHGFAILLIM:** Number of failures to train to a particular speed before that speed (and above) is removed from operation.<br>Used in conjunction with LTSSMERRSTS1. A value of 0 disables this feature. |
| 12 | 0x1<br>RW/P | **RPTSPDCHANGE:**<br>• 1: Allow speed changes to be logged as a bandwidth change in the LinkStatus register.<br>• 0: Don't report speed changes as a bandwidth change |
| 11 | 0x1<br>RW/P | **RPTCFGCHANGE:**<br>• 1: Allow width changes to be logged as a bandwidth change in the LinkStatus register.<br>• 0: Don't report width changes as a bandwidth change |
| 10 | 0x0<br>RW/V/P | **DISINITSPDCHNG:**<br>• 1: blocks initial speed change from Gen1-)Gen2/3 after DL_Active asserts<br>• 0: allows an automatic speed change attempt after reaching DL_Active to the highest common speed supported |
| 9 | 0x0<br>RW/P | **FORCEGEN3:**<br>• 1: Forces LTSSM to Detect and then trains link at 8GT/s mode<br>• 0: Releases forced speed |
| 8 | 0x0<br>RW/P | **FORCEGEN2:**<br>• 1: Forces LTSSM to Detect and then trains link at 5GT/s mode<br>• 0: Releases forced speed |
| 7 | 0x0<br>RW/P | **OVRDISGEN3:** Override of disabling gen3 |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6 | 0x0<br>RW/P | **OVRDISGEN2:** Override of disabling gen2 |
| 5 | 0x1<br>RW/P | **MAINTAINTLS:** Aggressively attempt to keep link at LinkCtl2.TLS speed setting. (DSP only) |
| 4 | 0x0<br>RW/P | **DISG3POLCFG:** Don't support advertising Gen3 speed in polling. |
| 3 | 0x0<br>RW/P | **DISG2POLCFG:** Don't support upper speed G2 in polling |
| 2 | 0x0<br>RW/P | **DISG3UNLESSRCVDTS:** Don't support speeds 8GT/s and higher until other device advertises it after a speed change failure or the 200ms timer expires. |
| 1 | 0x0<br>RW/P | **DISG2UNLESSRCVDTS:** Don't support speeds 5GT/s and higher until other device advertises it after a speed change failure or 200ms timer expires. |
| 0 | 0x1<br>RW/P | **DIRECTEDSPEEDCHNG:** Controls if the highest ever recorded speed supported is used to set the directed speed change internal variable for downstream operation.<br>• 0: two retrain cycles<br>• 1: one retrain cycle (DSP only) |

## 48.2.230 RCVRDETSTS—Offset B6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RO/V/P | **RCVRDETSTS:** The present receiver detect state of each lane in the cluster. |

## 48.2.231 RCVRERR—Offset B70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RW/P | **MASK:** prevent 8b/10b error from given cluster lane from being promoted to PCIE receiver error. Has no effect on RCVRERR.LANE |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7:0 | 0x0<br>RW/1C/V/P | **LANE:** Indication that a loggable receiver error occurred on a particular lane of the cluster. |

## 48.2.232 EBERR—Offset B74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RW/1C/V/P | **EB Full per lane (EBFULLERR):** Indication that a loggable receiver error occurred on a particular lane of the cluster. |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7:0 | 0x0<br>RW/1C/V/P | **EB Empty per lane (EBEMPTYERR):** Indication that a loggable receiver error occurred on a particular lane of the cluster. |

## 48.2.233 EBERRMSK—Offset B78h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B78h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RW/P | **MASK:** Prevent EB error from given cluster lane from being promoted to PCIE receiver error. Has no effect on EBERR. |

## 48.2.234 OBEINJCTL—Offset B7Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B7Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **Response function Done Signal Disable (RESPDONEDIS):** This bit will disable the done signal returned from the response logic.<br>• 0: Use the Done return signal to remove the assertion.<br>• 1: Disable the Done signal from altering the response function assertion. |
| 30:27 | 0x0<br>RW/P | **Lane select to inject error on (LNSLCT):**<br>• 0: Lane 0<br>• 1: Lane1<br>• ...<br>• 15: Lane 15<br>Assume N = max number of lanes for the port, then values greater than (N-1) are reserved. |
| 26:25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/P | **Inject Error in Symbols other than STP, END, SDP (OTHRINJ):**<br>• 0: Do not inject error on type Others.<br>• 1: Inject error on type Others when EINJEN is set. |
| 23 | 0x0<br>RW/P | **Inject Error in a SDP Symbol (SDPINJ):**<br>• 0: Do not inject error on SDP.<br>• 1: Inject error on SDP when EINJEN is set. |
| 22 | 0x0<br>RW/P | **Inject Error in a END Symbol (ENDINJ):**<br>• 0: Do not inject error on END.<br>• 1: Inject error on END when EINJEN is set. |
| 21 | 0x0<br>RW/P | **Inject Error in a STP Symbol (STPINJ):**<br>• 0: Do not inject error on STP.<br>• 1: Inject error on STP when EINJEN is set. |
| 20:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11:10 | 0x0<br>RW/P | **INJERRTYPE:** This field defines the type of error that is created.<br>In Gen2<br>• 00: 10b error, invert bit[0] of designated byte after 8b10b encoding<br>• 01: Disparity error on designated byte<br>• 10: 8b error, invert bit[0] of designated byte before 810b encoding<br>• 11: Reserved - no error injected<br>In Gen3<br>• 00: Invert bit[0] of designated Tx data byte<br>• 01: Invert syncheader[1:0] bits 01(=)10, changes a data block to a control block or vice versa<br>• 10: force syncheader[1:0]=2'b00 (invalid header)<br>• 11: force syncheader[1:0]=2'b11 (invalid header) |
| 9:6 | 0x0<br>RW/P | **BLKBYTEOFF:** In gen3, when OTHRINJ is set and the block type matches BLKTYPE, this field defines the Nth byte of block to place error (0 is 1st byte of block). |
| 5 | 0x0<br>RW/P | **HDRINJ:** In GEN3, inject an error on the block header that matched block type.<br>The error will be specified by encodings 01-11 of INJERRTYPE. |
| 4 | 0x0<br>RW/P | **BLKTYPE:** In GEN3, what block type to either inject a header error on to or qualify the block start with for offset error injection. |
| 3:2 | 0x0<br>RO | **RSVD2:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/P | **Error Injection Function Select (EINJFUNCTSEL):**<br>• 0: Select EINJ0 response function.<br>• 1: Select EINJ1 response function. |
| 0 | 0x0<br>RW/P | **Error Injection Enable (EINJEN):**<br>• 0: Disable error injection.<br>• 1: Enable error injection. |

## 48.2.235 LTSSMERRSTS0—Offset B80h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B80h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0 RW/1C/V/P | **Reserved** |
| 29 | 0x0 RW/1C/V/P | **EQMEVAL_TO:** Timeout in Phase2 (DSC) or Phase3 (USC) resulting in Rec.Speed |
| 28 | 0x0 RW/1C/V/P | **EQMEVALCK_TO:** Timeout in Phase2 (DSC) or Phase3 (USC) while locking on to next set of TS, may not result in Rec.Speed |
| 27 | 0x0 RW/1C/V/P | **EQSEVA_TO:** Timeout in Equalization Phase3 (DSC) or Phase2 (USC) resulting in Rec.Speed |
| 26 | 0x0 RW/1C/V/P | **EQSEVASYNC_TO:** Timeout in Equalization Phase3 (DSC) or Phase2 (USC) while locking on to next set of TS resulting in Rec.Speed |
| 25 | 0x0 RW/1C/V/P | **EQPH1_TO:** Timeout in Equalization Phase1 |
| 24 | 0x0 RW/1C/V/P | **EQPH0_TO:** Timeout in Equalization Phase0 |
| 23 | 0x0 RW/1C/V/P | **RECRCVRLCKWIDTH:** Recovery.RcvrLock Timeout Some TS to Config |
| 22 | 0x0 RW/1C/V/P | **RECRCVRLCKNOTALLTS:** Recovery.RcvrLock Timeout Some TS to Rec.RcvrCfg |
| 21 | 0x0 RW/1C/V/P | **RXL0SFTS_TO:** RxL0sFTS Timeout |
| 20 | 0x0 RW/1C/V/P | **UNEXPECTEDEI:** unexpected electrical idle event in L0 |
| 19 | 0x0 RW/1C/V/P | **LBMSTENTRY_TO:** Loopback.Master Entry timeout |
| 18 | 0x0 RW/1C/V/P | **DISWTEIOS_TO:** Disable.WaitForEIOS timeout |
| 17 | 0x0 RW/1C/V/P | **RECIDLEDOWN_TO:** Recovery.Idle Timeout link down |
| 16 | 0x0 RW/1C/V/P | **RECRCVRCFGSPDOWN_TO:** Recovery.RcvrConfigSpeed Timeout link down |
| 15 | 0x0 RW/1C/V/P | **RECRCVRCFGDOWN_TO:** Recovery.RcvrConfig Timeout link down |
| 14 | 0x0 RW/1C/V/P | **RECSPDEXITEIEX_TO:** Recovery.SpeedIdleExit Timeout waiting on EI exit |
| 13 | 0x0 RW/1C/V/P | **RECSPDIDLWTEI_TO:** Recovery.SpeedIdle Timeout waiting on EI |
| 12 | 0x0 RW/1C/V/P | **RECRCVRLCKBW_TO:** Recovery.RcvrLock Timeout to Recovery.Speed |
| 11 | 0x0 RW/1C/V/P | **RECRCVRLCKDOWN_TO:** Recovery.RcvrLock Timeout Link Down |
| 10 | 0x0 RW/1C/V/P | **UPL12RCVEIOS_TO:** Timed out in L12 Entry waiting on received EIOS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 9 | 0x0<br>RW/1C/V/P | **CFGIDLE_TO:** Timed out in Config.Idle |
| 8 | 0x0<br>RW/1C/V/P | **CFGCMPLT_TO:** Timed out in Config.Complete |
| 7 | 0x0<br>RW/1C/V/P | **CFGLNNUMAC_TO:** Timed out in Config.LaneNumAccept |
| 6 | 0x0<br>RW/1C/V/P | **CFGLNNUMWT_TO:** Timed out in Config.LaneNumWait |
| 5 | 0x0<br>RW/1C/V/P | **CFGLNKWDAC_TO:** Timed out in Config.LinkWidthAccept |
| 4 | 0x0<br>RW/1C/V/P | **CFGLNKWDST_TO:** Timed out in Config.LinkWidthStart |
| 3 | 0x0<br>RW/1C/V/P | **POLCFG_NOTS:** Timed out in Polling.Config w/out any TS seen |
| 2 | 0x0<br>RW/1C/V/P | **POLACT_NOTS:** Timed out in Polling.Active w/out any TS seen |
| 1 | 0x0<br>RW/1C/V/P | **PARTIAL_RCVRDET:** LTSSM detected something, but on 2nd look, no rcvr detected |
| 0 | 0x0<br>RW/1C/V/P | **SPEEDCHANGEFAIL:** A speed change attempt failed. |

## 48.2.236 LTSSMERRSTS1—Offset B84h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO/V/P | **LTERR0FERR:** Pointer to the first error logged in LTSSMERRSTS0, value is only valid if a bit is set in LTSSMERRSTS0 |
| 26:8 | 0x0<br>RW/1C/V/P | **Reserved** |
| 7 | 0x0<br>RW/1C/V/P | **G3FAILLIMERR:** Speed changes to 8GT/s have reached the maximum allowed failures, 8GT/s mode is now disabled.<br>See SPEEDCTL.SPDCHGFAILLIM register for more details. |
| 6:4 | 0x0<br>RO/V | **G3FAILCTR:**<br>The number of failed attempts to train to 8GT/s without successfully reaching L0 at 8GT/s.<br>This count resets to zero when G3FAILLIMERR is cleared or when L0 is reached at 8GT/s data rate without error. |
| 3 | 0x0<br>RW/1C/V/P | **G2FAILLIMERR:**<br>Speed changes to 5GT/s have reached the maximum allowed failures, 5GT/s mode is now disabled.<br>See SPEEDCTL.SPDCHGFAILLIM register for more details. |
| 2:0 | 0x0<br>RO/V | **G2FAILCTR:**<br>The number of failed attempts to train to 5GT/s without successfully reaching L0 at 5GT/s.<br>This count resets to zero when G2FAILLIMERR is cleared or when L0 is reached at 5GT/s data rate without error. |

## 48.2.237 EQEVALCTL6—Offset B8Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B8Ch

**Default:** 04904623h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28 | 0x0<br>RW/P | **REQLBMSTRCOEFF:** When intending to master loopback at Gen3 speeds, and entry is from the Config state, this bit can be used to enable the dictation of what far end Tx EQ settings to use.<br>The per lane settings are defined in the GEN3FARTXCOEFF#.D1/2/FTYPE fields.<br>The LNOVRD field is not required to be set.<br>Valid values must also be captured or programmed into the EQFARFSLF register.<br>This mode will not be enabled if loopback is entered from Recovery as it is assumed the values used to reach L0 in that scenario should be used. |
| 27:22 | 0x12<br>RW/P | **EIEREENERRCTRMSK:** Number of EIES cycles to wait before counting additional errors after EB/SOS error |
| 21:16 | 0x10<br>RW/P | **EIEREENTIME:** Number of EIES cycles to keep EIE reset bit de-asserted after EB/SOS error |
| 15 | 0x0<br>RW/V | **LOCBUSFETCHINIT:** Initiate a manual fetch self clears when complete. |
| 14 | 0x1<br>RW/P | **LOCBUSFETCHEN:** Enable Local bus fetch of coeff table. |
| 13 | 0x0<br>RW/P | **SPDCHGCTLEADPTEN:** Backup to the IOs doing CTLE for speed change to G3.<br>Control will initiate a CTLE adapt after speed changed and Rx EI Exit. |
| 12:10 | 0x1<br>RW/P | **LBCTLEADPTWT:** Time before starting CTLE after EI exit detect.<br>• 000 = 512ns<br>• 001 = 1us<br>• 010 = 1.5us<br>• 011 = 2us<br>• 100 = 4us<br>• 101 = 8us<br>• 110 = 12us<br>• 111 = 16us |
| 9 | 0x1<br>RW/P | **LBCTLEADPTEN:** Enable CTLE Adaption on entry to Loopback.Master @Gen3. |
| 8:6 | 0x0<br>RW/P | **REEQCTLEADPTWT:** Time before starting CTLE after entry to DP.PH1/UP.PH0.<br>• 000 = 512ns<br>• 001 = 1us<br>• 010 = 1.5us<br>• 011 = 2us<br>• 100 = 4us<br>• 101 = 8us<br>• 110 = 12us<br>• 111 = 16us |
| 5 | 0x1<br>RW/P | **REEQCTLEADPTEN:** Enable CTLE Adaption on entry to phase1/0 not from speed change w/EQTS2 |
| 4:0 | 0x3<br>RW/P | **BACKOFFTIME:** Time in EQ before starting CTLE Adapt and then TS lock val( 16 =) val[3:0],6'h0, CXLT_TOU_4NS 0-3.84us +0/-4ns val*256ns val)=16 =) val[4:0],5'h0, CXLT_TOU_128NS 0-126.98us +0/-128ns val*4us |

## 48.2.238 EQEVALCTL7—Offset B90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B90h

**Default:** 000000FFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0xff<br>RW/P | **FOMSTPLIM:** Max FOM value, if FOM)=FOMSTPLIM, exit evaluation. |

## 48.2.239 EQFARFSLF—Offset B9Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + B9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RO/V | **LFSAME:** All lanes of the port have equal LF |
| 16 | 0x0<br>RO/V | **FSSAME:** All lanes of the port have equal FS |
| 15:14 | 0x0<br>RO | **RSVD1:** Reserved |
| 13:8 | 0x0<br>RW/V/P | **FARLF:** This field shows the current EQ phase1 status of the link partners LF parameter.<br>The value is also used during entry to loopback master from config when requesting the use of coefficients.<br>If the link has not gone through phase1 of EQ, then software will need to manually write a value.<br>Anytime the link goes through phase1, this field is updated with the value advertised by the link partner (discarding any software written value). |
| 7:6 | 0x0<br>RO | **RSVD2:** Reserved |
| 5:0 | 0x0<br>RW/V/P | **FARFS:** This field shows the current EQ phase1 status of the link partners FS parameter.<br>The value is also used during entry to loopback master from config when requesting the use of coefficients.<br>If the link has not gone through phase1 of EQ, then software will need to manually write a value.<br>Anytime the link goes through phase1, this field is updated with the value advertised by the link partner (discarding any software written value). |

## 48.2.240 FOMINCCTL—Offset BA4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BA4h

**Default:** 03046820h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x3 RW/P | **SlowTimerThresh:** FOM feedback value threshold to switch to Slow timer from fast |
| 23 | 0x0 RW/P | **ErrMaskEn:** Enable Error Mask window |
| 22 | 0x0 RW/P | **Tem_Unit:** Error mask window time unit<br>• 0: 28ns + val* 32ns upto 2044ns (+0/-4ns)<br>• 1: 896ns + val* 1024ns upto 65408ns (+0/-128ns) |
| 21:16 | 0x4 RW/P | **Tem_Val:** Error mask window time count |
| 15:14 | 0x1 RW/P | **Tsi_Unit:** Slow Time Unit |
| 13:8 | 0x28 RW/P | **Tsi_Val:** Slow Time Count Value |
| 7:6 | 0x0 RW/P | **Tfi_Unit:** Fast Time Unit<br>• 00: 28ns + val* 32ns upto 2044ns (+0/-4ns)<br>• 01: 896ns + val* 1024ns upto 65408ns (+0/-128ns)<br>• 10: 7us + val* 8us upto 504us (+0/-1us)<br>• 11: 48us + val* 64us upto 4.08ms (+0/-16us) |
| 5:0 | 0x20 RW/P | **Tfi_Val:** Fast Time Count Value |

## 48.2.241 PHYRECAL—Offset BA8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BA8h

**Default:** 0000000Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0 RO | **RSVD0:** Reserved |
| 3 | 0x1 RW/P | **G3RecOnErrEn:** Request recal on Gen3 Recovery due to Error |
| 2 | 0x1 RW/P | **G2RecOnErrEn:** Request recal on Gen2 Recovery due to Error |
| 1 | 0x0 RW/P | **G3L1ExitEn:** Request recal on Gen3 L1 exit |
| 0 | 0x0 RW/P | **G2L1ExitEn:** Request recal on Gen2 L1 exit |

## 48.2.242 PLLSTBYCTL—Offset BACh

This register is only valid for project specific w/ PLLSTBY feature.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BACh

**Default:** 00000100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **PLL Standby In Progress Assertion (PSIP):** When PLL standby is enable gives a range of time to all idle to L1 entry. This indicates to the PCI PLL Control block to assert PSIP. Also known as t200. Range : 0 = 70ns 1 = 150ns 2 = 230ns ... 15= 1270ns<br><br>**Power Well:** PRST |
| 27:24 | 0x0<br>RW | **PLL Standby Assertion (PSTBYA):** Time window to allow for synchronization between idle indications and PLL control logic. This window starts after L1 entry and PLL standby Enabled. Also known as t200. Range: 0 = 40ns 1 = 50ns 2 = 60ns ... 15= 190ns<br><br>**Power Well:** PRST |
| 23:20 | 0x0<br>RW | **PLL Standby Min Assertion time (PSMAT):** Minimum time that the PLL Control block can put the PLL in standby. Anytime shorter than the min time can cause indeterminate PLL behavior. Also known as t202. Range: 0 = 150ns 1 = 310ns 2 = 470ns ... 15= 2550ns<br><br>**Power Well:** PRST |
| 19:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x1<br>RW | **P2 vs P1/2 for PLL Idle (P2VSP1):** 0 = lane in P1 equal to idle for PLL standby (see PIPECTL for P2 disable) 1 = lane in P2 equal to idle for PLL standby<br><br>**Power Well:** PRST |
| 7:4 | 0x0<br>RO | **RSVD1:** Reserved |
| 3:0 | 0x0<br>RW | **PLL Standby Enable (PSE):** A one enables the PLL control block to put PLL1-4 in standby when the port is in L1 and all idle conditions are met.<br><br>**Power Well:** PRST |

## 48.2.243 PLUTXEQCTL—Offset BB0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BB0h

**Default:** 078A0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:22 | 0x1e<br>RW/V/P | **8GT/s Full Swing (FS):** This field is used for setting 8GT/s full swing coefficient parameter. |
| 21:16 | 0xa<br>RW/V/P | **8GT/s Low Frequency (LF):** This field is used for setting 8GT/s low frequency coefficient parameter. |
| 15:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 48.2.244 PHYCGIDLEMSK—Offset BB4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BB4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RW/P | **PLL_L1IDLE:** Disables placing the PLL in standby during L1.Idle state This field is only valid for project specific w/ PLLSTBY feature. |
| 4 | 0x0<br>RW/P | **PLL_L2IDLE:** Disables placing the PLL in standby during L2.Idle state This field is only valid for project specific w/ PLLSTBY feature. |
| 3 | 0x0<br>RW/P | **PLL_DISSUCC:** Disables placing the PLL in standby during Disable.Success state This field is only valid for project specific w/ PLLSTBY feature. |
| 2 | 0x0<br>RW/P | **Disable L1 Clock Gating (L1IDLE):** Disables clock gating in L1.Idle state |
| 1 | 0x0<br>RW/P | **Disable L2 Clock Gating (L2IDLE):** Disables clock gating in L2.Idle state |
| 0 | 0x0<br>RW/P | **Disable LTSSM Disable State Clock Gating (DISSUCC):** Disables clock gating in Disable.Success state |

## 48.2.245 EQEVALCTL4—Offset BB8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BB8h

**Default:** 64320000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **REVERSEEQTS2:** Allows EQTS2s to be sent by upstream port or decoded by downstream port to override the LinkEqCtl.DownStreamPortTxPreset value.<br>For an upstream port, the PREFPRESET1 value will be sent in the EQTS2s it sends to the upstream component. For a downstream port, this will enable it to accept EQTS2s from the upstream port and override the LinkEqCtl.DownstreamTxPreset value. |
| 30:24 | 0x64<br>RW/P | **QCHKTMVAL:** Final evaluation loop time.<br>This loop will not ask the IOs to enabled any adaptive eq logic and is used as a quick sanity check of link errors. |
| 23 | 0x0<br>RW/P | **QCHKTMUNIT:** 0 = time limit is QCHKTMVAL * 1us, 1 = time limit is QCHKTMVAL * 128us |
| 22:20 | 0x3<br>RW/P | **HINTIDXMODE:** Hint mode index lookup FarTxWeight (FTW) = Using TxPreset captured in Phase1, apply mapping from UPHINTMAP0/1 register LocTxWeight (LTW) = Using TxPreset captured in EQTS2s (LaneEqCtl.UPTP), apply mapping from UPHINTMAP0/1 register RxHintWeight (RHW) = Using Rx Hint captured in EQTS2s (LaneEqCtl.UPRPH), apply mapping from UPHINTMAP2 register Register Value: Hint Index Result<br>• 000: (LTW+FTW)/2<br>• 001: FTW<br>• 010: LTW<br>• 100: RHW<br>• 011: (LTW+FTW)/2<br>• 101: (RHW+FTW)/2<br>• 110: (RHW+LTW)/2<br>• 111: (RHW+LTW+FTW)/3<br>The resultant index is used in the comparison of UPHISELFOMTBL.<br>UPHISELFOMTBL.MST If the value calculated according to EQEVALCTL4.HINTIDXMODE is less than the programmed value, treat channel as short and use FOM stage1 entries from the FOM Hint Table.<br>UPHISELFOMTBL.MLT If the value calculated according to EQEVALCTL4.HINTIDXMODE is less than the programmed value and greater or equal to MST, treat the channel as medium-short and use UPHIFOMTBLM0-7.MSD1,D2,FT for FOM Stage1 entries.<br>UPHISELFOMTBL.MLT If the value calculated according to EQEVALCTL4.HINTIDXMODE is less than the programmed value and greater or equal to MST, treat the channel as medium-short and use UPHIFOMTBLM0-7.MSD1,D2,FT for FOM Stage1 entries.<br>UPHISELFOMTBL.LT If the value calculated according to EQEVALCTL4.HINTIDXMODE is less than the programmed value and greater or equal to MLT, treat the channel as medium-long and use UPHIFOMTBLM0-7.MLD1,D2,FT for FOM Stage1 entries.<br>If the value calculated according to EQEVALCTL4.HINTIDXMODE is greater or equal to the programmed value, treat the channel as long and use UPHIFOMTBLL0-7.LD1,D2,FT for FOM Stage1 entries. |
| 19 | 0x0<br>RW/V/P | **RXHINT7OVR_EN:** When set to 1, the starting coeff to evaluate (phase2 upstream, phase3 downstream) will be derived from Rx Hint Table entry DBXX (7).<br>This overrides the preferred preset mode and the normal Rx Hint table mode. |
| 18 | 0x0<br>RW/P | **RXHINTCOEFF_EN:** If this field is set and FOMSEARH_EN=1 and port is an Upstream port, use HINTIDXMODE and UPHISELFOMTBL to select FOM Stage1 points<br>-OR- FOMSEARCH_EN=1 and port is a Downstream port, field is unused (except if EQEVALCTL2.EVALATT==1).<br>-OR- FOMSEARCH_EN=0, use HINTIDXMODE result to index into FOMHINTTBL for single coefficient evaluation. |
| 17 | 0x1<br>RW/V/P | **FOMSEARCH_EN:** Figure of Merit search enable. |
| 16:10 | 0x0<br>RO | **RSVD0:** Reserved |
| 9:0 | 0x0<br>RW/P | **FOMERRTHRESH:** Figure of Merit error threshold |

## 48.2.246 EQEVALCTL5—Offset BBCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BBCh

**Default:** 3A87F003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **RSVD0:** Reserved |
| 30:27 | 0x7<br>RW/P | **PhSTimerVal:** Slave phase (UP=Phase3, DP=Phase2) timeout. time limit is (val+1)*4ms. |
| 26:23 | 0x5<br>RW/P | **PhMTimerVal:** Master phase (UP=Phase2, DP=Phase3) timeout. time limit is (val+1)*4ms. |
| 22:20 | 0x0<br>RW/P | **PhaseTODis:** Disable phase 1/2/3 timeouts bit 0 - phase 1 bit 1 - phase 2 bit 3 - phase 3 |
| 19:12 | 0x7f<br>RW/P | **USEPREEVALWAIT:** Time to wait after lock (or lock timeout) before enabling the IOs to evaluate the new tx preset settings.<br>This allows the circuits some extra time to setup their evaluation logic.<br>Time is value*4us. |
| 11:4 | 0x0<br>RW/P | **COEFFEVALWAIT:** Time to wait after lock (or lock timeout) before enabling the IOs to evaluate the new coeff settings.<br>This allows the circuits some extra time to setup their evaluation logic.<br>Time is value*4us. |
| 3:0 | 0x3<br>RW/P | **LOCKTIME:** Max time to allow block lock on new coeff request.<br>Time is value*4us.<br>When this field is set to 0, it will remain in Lock state until lock is achieved or the primary timer for Phase2/3 expires (thus an EQ failure). |

## 48.2.247 LEKBLNERRCNT0—Offset BC0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BC0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **RSVD0:** Reserved |
| 29:25 | 0x0<br>RO/V | **Lane 5 Error Count (ERRCNT5):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 24:20 | 0x0<br>RO/V | **Lane 4 Error Count (ERRCNT4):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 19:15 | 0x0<br>RO/V | **Lane 3 Error Count (ERRCNT3):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 14:10 | 0x0<br>RO/V | **Lane 2 Error Count (ERRCNT2):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 9:5 | 0x0<br>RO/V | **Lane 1 Error Count (ERRCNT1):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 4:0 | 0x0<br>RO/V | **Lane 0 Error Count (ERRCNT0):** Error count seen on a lane for the current time window.<br>Decrements progressively to 0 if no errors seen over a prolonged time period. |

## 48.2.248 LEKBLNERRCNT1—Offset BC4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BC4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **RSVD0:** Reserved |
| 9:5 | 0x0<br>RO/V | **Lane 7 Error Count (ERRCNT7):** Error count seen on a lane for the current time window. Decrements progressively to 0 if no errors seen over a prolonged time period. |
| 4:0 | 0x0<br>RO/V | **Lane 6 Error Count (ERRCNT6):** Error count seen on a lane for the current time window. Decrements progressively to 0 if no errors seen over a prolonged time period. |

## 48.2.249 EQEVALCTL—Offset BD0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BD0h

**Default:** 00080808h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **RSVD0:** Reserved |
| 30 | 0x0<br>RW/P | **BYPASSEQ:** Bypass all aspects of Adaptive EQ. |
| 29 | 0x0<br>RW/P | **NODLINITBLK:** Don't block the DL state machine from starting DL Init once initial L0 reached. |
| 28 | 0x0<br>RW/P | **BYPASSEQTS2:** For a downstream link, this allows entry to Gen3 w/out sending EQTS2 in 8b/10b mode prior to transmitting at 8GT/s. |
| 27:24 | 0x0<br>RW/P | **DEFPRESET:** Preset that will be used if 8GT/s entered w/out receiving EQTS2s. preset 0000 is -6dB de-emphasis and 0dB preshoot. |
| 23 | 0x0<br>RW/P | **FLIPCOEFFUPDPOL:** Chicken bit to flip what coeff up/down means to core. |
| 22:16 | 0x8<br>RW/P | **PREACCTHR:** Pre cursor eval threshold. The threshold used is value*8 for calculation. |
| 15 | 0x0<br>RO | **RSVD1:** Reserved |
| 14:8 | 0x8<br>RW/P | **MAINACCTHR:** Main cursor eval threshold. The threshold used is value*8 for calculation. |
| 7 | 0x0<br>RO | **RSVD2:** Reserved |
| 6:0 | 0x8<br>RW/P | **POSTACCTHR:** Post cursor eval threshold. The threshold used is value*8 for calculation. |

## 48.2.250 EQEVALCTL2—Offset BD4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BD4h

**Default:** 0000FF55h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/P | **BLKUPEQREQ:** When this bit is set to 1, the following two events will NOT set the request eq bit in TS2s during Rec.RcvrCfg following Rec.Equalization.<br>• Equalization engine deems link usable but not optimal (this would normally trigger an immediate eq request in TS2s)<br>• Coeff received in Rec.RcvrLck follow Rec.Eq don't match negotiated coeff during Eq. |
| 30 | 0x0<br>RW/P | **BLKUPSPEED:** H/W will not initiate an up step in speed to 8GT/s in order to advertise or re-do equalization |
| 29 | 0x0<br>RW/P | **IGNOREQUIESCE:** H/W will ignore quiesce bit received from downstream component |
| 28 | 0x0<br>RW/P | **HWQUIESCE:** H/W will require quiescing of the link prior to re-equalization |
| 27 | 0x0<br>RW/P | **HWREEQEN:** Allow HW to initiate re-equalization. |
| 26 | 0x0<br>RW/P | **EXTEIEOS:** Enable extended EIEOS request during master eval stage (up component phase 3, down component phase 2). |
| 25 | 0x0<br>RW/P | **IGNOREMD:** Echo coeff back but don't change them to the IO. |
| 24 | 0x0<br>RW/P | **SKIPEVALPH:** Skip execution of the master eval coeff phase (up component phase 3, down component phase 2).<br>For downstream it skips phase2/3 completely per PCIe Spec.<br>For upstream it passes quickly through phase2 doing no real eval. |
| 23 | 0x0<br>RW/P | **REEQSPDDEGRADE:** When set, a self detected need to re-equalize will result in a down shift in speed and a disabling of the current rate (8GT/s or higher) and all rates above it. |
| 22:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:8 | 0xff<br>RW/V/P | **EVALATT:** Max number of coeff for a lane to eval before quitting eval phase.<br>Value of 0 means no limit.<br>The value is scaled by the link partners advertised FS value (adj_val = roundup(val*(FS+1)/64)).<br>If FS is 63, the spec max, the value would be unaltered, whereas for a FS of 31, the value written to the config register would be halved.<br>One attempt will be made if the programmed value is greater than 0 and the adjusted value would have resulted in a value of 0.<br>Any evaluation attempt following an InvalidRequest indication on the PIPE interface is not counted as an attempt. |
| 7:0 | 0x55<br>RW/P | **EVALTIME:** Field defined to do each coeff eval as following.<br>Evaltime[7:6] = unit Evaltime[5:0] = value Unit:<br>• 11=)1ms<br>• 10=)128us<br>• 01=)16u<br>• 00=)1us<br>This field should not be programmed in value of 0 (unit=any, value=0) for devices connected to WM design. |

## 48.2.251 EQEVALCTL3—Offset BD8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BD8h

**Default:** 33F30802h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x3<br>RW/P | **REPEATLIM:** Maximum number of times a previously tried coeff can be attempted.<br>A value of 0 means disable the repeat check.<br>Repeats are defined as coeff already seen in the history matrix.<br>Each time a coeff lookup in the history matrix returns a hit and the limit not reached, the history matrix is reset and the repeat counter incremented. |
| 27:24 | 0x3<br>RW/P | **INVREQATTLIM:** Maximum number of repeated times the MAC will allow the PIPE-Phy to request an out-of-bounds coeff (MAC responds to each by asserting InvalidRequest) before aborting the equalization evaluation process.<br>The PIPE3 spec gives no bounds, but if after some number of attempts, the PIPE-Phy keeps requesting the same invalid update, it makes no sense to continue.<br>A value of 0 disables this check. |
| 23 | 0x1<br>RW/P | **AllowFromMainOnly:** When set to 1, the main cursor is the only cursor pre or post can get a unit from.<br>So if main is at its minimum value, then an addition to post (or pre) can only occur if a subtraction from pre (or post) is also requested.<br>This prevents adjusting the cursor ratio of pre or post in the opposite direction than is desired.<br>When set to 0, a pre/post edit will take a unit from post/pre if the main cursor does not have a unit to give. |
| 22 | 0x1<br>RW/P | **OptCursorEdit:** If pre and post cursor movement requests are in opposite directions, service both at once if legally possible.<br>If not set, it will only edit 1 cursor using the priority defined by PreVsPostRR and PreOverPost |
| 21 | 0x1<br>RW/P | **PreVsPostRR:** Edits to pre vs post cursor will be done in a fair round robin manner.<br>The PreOverPost field will dictate the first cursor to be modified.<br>If not set, it will follow a strict priority to pre or post based on PreOverPost field. |
| 20 | 0x1<br>RW/P | **PreOverPost:** Pre-cursor adjustments will be taken over post-cursor adjustments when set to 1, else post has priority. |
| 19:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x1<br>RW/P | **UNDERT2REDO:** If block error count under Threshold 2 but equal to or over Threshold 1, request a redo of equalization |
| 16 | 0x1<br>RW/P | **OVERT2SPDFAIL:** If block error count measured over Threshold 2, force link to fail down to lower speed |
| 15:8 | 0x8<br>RW/P | **ERRTHRESH2:** Block error count limit considered usable, but may spawn a redo of Equalization if enabled.<br>Setting of 0 causes all checks to at least pass the Threshold 2 check. |
| 7:0 | 0x2<br>RW/P | **ERRTHRESH1:** Block error count limit considered optimal.<br>Setting of 0 disables block error check. |

## 48.2.252 EQEVALSTS—Offset BDCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BDCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RO/V/P | **Lane 7 at EVAL Loop (LNLOOP7):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 22 | 0x0<br>RO/V/P | **Lane 6 at EVAL Loop (LNLOOP6):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 21 | 0x0<br>RO/V/P | **Lane 5 at EVAL Loop (LNLOOP5):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 20 | 0x0<br>RO/V/P | **Lane 4 at EVAL Loop (LNLOOP4):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 19 | 0x0<br>RO/V/P | **Lane 3 at EVAL Loop (LNLOOP3):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 18 | 0x0<br>RO/V/P | **Lane 2 at EVAL Loop (LNLOOP2):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 17 | 0x0<br>RO/V/P | **Lane 1 at EVAL Loop (LNLOOP1):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 16 | 0x0<br>RO/V/P | **Lane 0 at EVAL Loop (LNLOOP0):**<br>FBDIR = Lane hit Loop Limit: Lane evaluation stopped due to repeated coeff (IO tap update resulted in a loop)<br>FOM = Lane in Paused state: PAUSE gets set when lane SM goes to PAUSE |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RO/V/P | **Lane 7 at optimal EQ (LNOPT7):** Lane is running at optimal EQ setting |
| 6 | 0x0<br>RO/V/P | **Lane 6 at optimal EQ (LNOPT6):** Lane is running at optimal EQ setting |
| 5 | 0x0<br>RO/V/P | **Lane 5 at optimal EQ (LNOPT5):** Lane is running at optimal EQ setting |
| 4 | 0x0<br>RO/V/P | **Lane 4 at optimal EQ (LNOPT4):** Lane is running at optimal EQ setting |
| 3 | 0x0<br>RO/V/P | **Lane 3 at optimal EQ (LNOPT3):** Lane is running at optimal EQ setting |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RO/V/P | **Lane 2 at optimal EQ (LNOPT2):** Lane is running at optimal EQ setting |
| 1 | 0x0<br>RO/V/P | **Lane 1 at optimal EQ (LNOPT1):** Lane is running at optimal EQ setting |
| 0 | 0x0<br>RO/V/P | **Lane 0 at optimal EQ (LNOPT0):** Lane is running at optimal EQ setting |

## 48.2.253 EQEVALSTS2—Offset BE0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BE0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **RSVD0:** Reserved |
| 8 | 0x0<br>RW/1C/V | **REEQSPDDISG3:** When set, hardware has disabled 8GT/s and above data rates due to a need to re-equalize. |
| 7:0 | 0x0<br>RO/V/P | **EVALATTS:** Number of eval attempts to get all lanes optimal. |

## 48.2.254 EQEVALSTS3—Offset BE4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BE4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23 | 0x0<br>RO/V/P | **Lane 7 at EVAL Rule Violation (LNRULELIM7):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 22 | 0x0<br>RO/V/P | **Lane 6 at EVAL Rule Violation (LNRULELIM6):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 21 | 0x0<br>RO/V/P | **Lane 5 at EVAL Rule Violation (LNRULELIM5):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 20 | 0x0<br>RO/V/P | **Lane 4 at EVAL Rule Violation (LNRULELIM4):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 19 | 0x0<br>RO/V/P | **Lane 3 at EVAL Rule Violation (LNRULELIM3):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 18 | 0x0<br>RO/V/P | **Lane 2 at EVAL Rule Violation (LNRULELIM2):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 17 | 0x0<br>RO/V/P | **Lane 1 at EVAL Rule Violation (LNRULELIM1):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 16 | 0x0<br>RO/V/P | **Lane 0 at EVAL Rule Violation (LNRULELIM0):**<br>FBDIR = Hit invalid request too many times: Lane evaluation stopped due to coeff rule boundary (IO tap update would result in rule violation)<br>FOM = Search complete: Complete gets set when lane SM goes to OPTIMAL or NON_OPTIMAL and the full search has completed (so cases of non-optimal w/out complete being set are possible). |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RO/V/P | **Lane 7 Met Error Threshold 2 (LNERRTH2MET7):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 6 | 0x0<br>RO/V/P | **Lane 6 Met Error Threshold 2 (LNERRTH2MET6):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 5 | 0x0<br>RO/V/P | **Lane 5 Met Error Threshold 2 (LNERRTH2MET5):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RO/V/P | **Lane 4 Met Error Threshold 2 (LNERRTH2MET4):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 3 | 0x0<br>RO/V/P | **Lane 3 Met Error Threshold 2 (LNERRTH2MET3):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 2 | 0x0<br>RO/V/P | **Lane 2 Met Error Threshold 2 (LNERRTH2MET2):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 1 | 0x0<br>RO/V/P | **Lane 1 Met Error Threshold 2 (LNERRTH2MET1):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |
| 0 | 0x0<br>RO/V/P | **Lane 0 Met Error Threshold 2 (LNERRTH2MET0):** Non-optimal lane met error threshold 2 requirements from EQEVALCTL3.ERRTHRESH2 |

## 48.2.255 G3FRAMEERR—Offset BECh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:9-12, 14-17, F:0] + BECh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **RSVD0:** Reserved |
| 30:27 | 0x0<br>RO/V/P | **G3FRAMEFERR:** First error pointer to bits 26:0 of this register |
| 26:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11 | 0x0<br>RW/1C/V/P | **FCRC:** TLP framing crc error |
| 10 | 0x0<br>RW/1C/V/P | **ITKNL0:** Invalid token on lane 0 (may include misplaced legal token) |
| 9 | 0x0<br>RW/1C/V/P | **ITKNLI:** Invalid token post logical idle (may include misplaced legal token) |
| 8 | 0x0<br>RW/1C/V/P | **ITKNDL:** Invalid token post DLLP (may include misplaced legal token) [as x1, means in DLLP token too] |
| 7 | 0x0<br>RW/1C/V/P | **ITKNTL:** Invalid token post TLP (may include misplaced legal token) [as x1/x2, includes bad EDB token] |
| 6 | 0x0<br>RW/1C/V/P | **SDSFE:** bad/unexpected data following SDS block |
| 5 | 0x0<br>RW/1C/V/P | **BKFE:** block framing error (header mismatch, bad header, etc.) |
| 4 | 0x0<br>RW/1C/V/P | **BK1FE:** unexpected type 1 (OS) header |
| 3 | 0x0<br>RW/1C/V/P | **EDS:** bad data following EDS (includes EDS not at end of block) |
| 2 | 0x0<br>RW/1C/V/P | **STP2:** two TLP start tokens in same symbol time |
| 1 | 0x0<br>RW/1C/V/P | **SDP2:** two DLLP start tokens in same symbol time |
| 0 | 0x0<br>RW/1C/V/P | **RDOK:** rcvd data ok removed unexpectedly (rcvr. imp. specific) |

## 48.3 Sideband Registers - Devices 9-12

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 48.3.1 PPBIFCTL_PRIV—Offset 10h

Port Configuration Bifurcation Control for MLW (x8) vs Port number

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xB4] + 10h
**Host Memory Space:** SBREG_BAR + 0xB40000 + 10h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits of PPBIFCTL register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit. |
| 30:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RW/V/L | **Bifurcation Control 0 (BIFCTL0):**<br>• 111-101: Reserved<br>• 100: x8<br>• 011: x4x4<br>• 010: x4x2x2<br>• 001: x2x2x4<br>• 000: x2x2x2x2<br>A write of '1' on the CL will lock this register field down as bifurcation control status information. |

## 48.4 Sideband Registers - Devices 14-17

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 48.4.1 PPBIFCTL_PRIV—Offset 10h

Port Configuration Bifurcation Control for MLW (x8) vs Port number

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xB3] + 0h
**Host Memory Space:** SBREG_BAR + 0xB30000 + 10h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits of PPBIFCTL register only.<br>• 1b: Lock<br>• 0b: Unlocked<br>This bit is self-locking. Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no affect on this bit. |
| 30:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2:0 | 0x0<br>RW/V/L | **Bifurcation Control 0 (BIFCTL0):**<br>• 111-101: Reserved<br>• 100: x8<br>• 011: x4x4<br>• 010: x4x2x2<br>• 001: x2x2x4<br>• 000: x2x2x2x2<br>A write of '1' on the CL will lock this register field down as bifurcation control status information. |

§ §

# 49 SMBus - Host - B0, D18, F0

## 49.1 Introduction and Index

The host-accessible registers for the SMBus - Host are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 49.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 18 (decimal), Function 0. The offset addresses are listed.

**Table 49-1. Summary of PCI Configuration Registers—0/18/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "VID—Offset 0h" on page 1820 |
| 2 | 2 | 19AC | "DID—Offset 2h" on page 1820 |
| 4 | 2 | 0000 | "PCICMD—Offset 4h" on page 1821 |
| 6 | 2 | 0010 | "PCISTS—Offset 6h" on page 1822 |
| 8 | 1 | 00 | "RID—Offset 8h" on page 1823 |
| 9 | 3 | 00088000 | "CCR—Offset 9h" on page 1823 |
| 10 | 8 | 0000000000000004 | "SMTBAR—Offset 10h" on page 1824 |
| 2C | 2 | 8086 | "SVID—Offset 2Ch" on page 1824 |
| 2E | 2 | 0000 | "SID—Offset 2Eh" on page 1825 |
| 3C | 1 | 00 | "INTL—Offset 3Ch" on page 1825 |
| 3D | 3 | 01 | "INTP—Offset 3Dh" on page 1826 |
| 40 | 2 | 8010 | "EXPCAPLST—Offset 40h" on page 1826 |
| 44 | 4 | 10008001 | "DEVCAP—Offset 44h" on page 1827 |
| 48 | 2 | 0000 | "DEVCTL—Offset 48h" on page 1829 |
| 4A | 2 | 0000 | "DEVSTS—Offset 4Ah" on page 1830 |
| 80 | 2 | 8C01 | "PMCAPLST—Offset 80h" on page 1831 |
| 8C | 2 | 0005 | "MSICAPLST—Offset 8Ch" on page 1831 |
| 8E | 2 | 0180 | "MSICTL—Offset 8Eh" on page 1832 |
| 90 | 8 | 0000000000000000 | "MSIADDR—Offset 90h" on page 1832 |
| 98 | 4 | 00000000 | "MSIDATA—Offset 98h" on page 1833 |
| 9C | 4 | 00000000 | "MSIMSK—Offset 9Ch" on page 1834 |
| EA | 2 | 0000 | "PLKCTL—Offset EAh" on page 1834 |

## 49.1.2 Host Memory Space—SMTBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 49-2. Summary of Memory Mapped I/O Registers—SMTBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "GCTRL—Offset 0h" on page 1835 |
| 8 | 8 | 0000000000000000 | "SMTICL—Offset 8h" on page 1836 |
| 10 | 4 | 00000000 | "ERRINTMSK—Offset 10h" on page 1837 |
| 14 | 4 | 00000000 | "ERRAERMSK—Offset 14h" on page 1838 |
| 18 | 4 | 00000000 | "ERRSTS—Offset 18h" on page 1839 |
| 1C | 4 | 00000000 | "ERRINFO—Offset 1Ch" on page 1840 |
| 100 | 8 | 0000000000000000 | "MDBA—Offset 100h" on page 1840 |
| 108 | 4 | 00000000 | "MCTRL—Offset 108h" on page 1841 |
| 10C | 4 | 00000000 | "MSTS—Offset 10Ch" on page 1842 |
| 110 | 4 | 00000000 | "MDS—Offset 110h" on page 1843 |
| 114 | 4 | 00061020 | "RPOLICY—Offset 114h" on page 1844 |
| 200 | 8 | 0000000000000000 | "TBBA—Offset 200h" on page 1846 |
| 208 | 4 | 00000010 | "TCTRL—Offset 208h" on page 1847 |
| 20C | 4 | 00000000 | "TSTS—Offset 20Ch" on page 1848 |
| 210 | 4 | 00000000 | "TBS—Offset 210h" on page 1849 |
| 218 | 4 | 00000000 | "HTHP—Offset 218h" on page 1849 |
| 21C | 4 | 00000000 | "FTTP—Offset 21Ch" on page 1850 |
| 220 | 4 | 00000058 | "TRxCTRL—Offset 220h" on page 1850 |
| 224 | 4 | 00000000 | "TRxSTS—Offset 224h" on page 1851 |
| 228 | 4 | 00000000 | "TACTRL—Offset 228h" on page 1852 |
| 22C | 4 | 00000000 | "TPOLICY—Offset 22Ch" on page 1853 |
| 240 | 4 | 00000000 | "GPBRCTRL—Offset 240h" on page 1855 |
| 244 | 4 | 00000000 | "GPBRDBUF—Offset 244h" on page 1856 |
| 280 | 4 | 00000000 | "SMTARPCTRL—Offset 280h" on page 1857 |
| 290 | 8 | 8086000000000000 | "UDID0—Offset 290h" on page 1858 |
| 298 | 8 | 8108808600000024 | "UUDID0—Offset 298h" on page 1859 |
| 2A0 | 8 | 8086000000000000 | "UDID1—Offset 2A0h" on page 1860 |
| 2A8 | 8 | 8108808600000024 | "UUDID1—Offset 2A8h" on page 1861 |
| 300 | 4 | 00000005 | "SPGT—Offset 300h" on page 1862 |
| 304 | 4 | 08080000 | "SPMT—Offset 304h" on page 1863 |
| 308 | 4 | 00000000 | "SPST—Offset 308h" on page 1864 |
| 30C | 4 | 00002024 | "SMBFT—Offset 30Ch" on page 1865 |
| 310 | 4 | 000015E1 | "CLTC—Offset 310h" on page 1866 |
| 380 | 4 | 80000000 | "DCLKGT—Offset 380h" on page 1866 |
| 384 | 4 | 00000011 | "SUSCHKB—Offset 384h" on page 1867 |
| 388 | 4 | 00000003 | "DBCTRL—Offset 388h" on page 1868 |
| 38C | 4 | 00000300 | "DBSTS—Offset 38Ch" on page 1869 |

## 49.2 Registers in Configuration Space

### 49.2.1 VID—Offset 0h

The Vendor ID Register identifies this device as an Intel device and adheres to the definitions in the PCI Local Bus Specification, Revision 3.0.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This is the standard 16-bit value assigned to Intel by PCI-SIG. |

### 49.2.2 DID—Offset 2h

The Device ID Register identifies this device and adheres to the PCI Local Bus Specification, Revision 3.0.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 2h

**Default:** 19ACh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x19ac<br>RO/V | **Device ID (DID):** This is the SMT Device identification Number. |

## 49.2.3 PCICMD—Offset 4h

The PCI Command Register follows a subset of the PCI Local Bus Specification, Revision 3.0. This register provides the basic control of the ability of the device to initiate and respond to transactions sent to it and maintains compatibility with PCI configuration space.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (INTxD):** Controls the ability to generate legacy INTx interrupt messages. When set, functions are prevented from asserting INTx interrupt messages. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBE):** Hard-wired to 0. |
| 8 | 0x0<br>RW | **SERR# Enable (SEE):** This bit indicates whether the device is allowed to signal a PCIe SERR condition.<br>• 0: The device is disabled from generating fatal/non-fatal errors.<br>• 1: The device is enabled to send fatal/non-fatal errors. |
| 7 | 0x0<br>RO | **RSVD:** Reserved |
| 6 | 0x0<br>RW | **Parity Error Response (PERE):** This bit controls the setting of the master data parity error bit in the Status Register in response to a parity error received on the PCI Express interface (either internal queue errors or a poisoned TLP).<br>• 0: Parity errors are ignored by the device.<br>• 1: The device reports parity errors. |
| 5 | 0x0<br>RO | **RSVD:** Reserved |
| 4 | 0x0<br>RO | **RSVD:** Reserved |
| 3 | 0x0<br>RO | **RSVD:** Reserved |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the ability of this device to initiate transactions to memory including MMIO.<br>• 0: The function is not allowed to issue any memory requests.<br>• 1: The function is allowed to issue memory requests. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls whether the device responds to memory space transactions as located by its memory space BAR.<br>• 0: Disables device's memory BAR to be decoded as a valid target address.<br>• 1: Enables device's memory BAR to be decoded as a valid target address. |
| 0 | 0x0<br>RO | **RSVD:** Reserved |

## 49.2.4 PCISTS—Offset 6h

The PCI Status Register follows a subset of the PCI Local Bus Specification, Revision 3.0. This register maintains compatibility with PCI configuration space. Since this register is part of the standard PCI header, there is a PCISTS register per PCI function.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Detected Parity Error (DPE):** This bit is set when the device receives an uncorrectable data error or Address/Control parity error regardless of the Parity Error Response bit in the "PCICMD—Offset 4h" on page 1821. |
| 14 | 0x0<br>RW/1C/V | **SERR# Asserted (SSE):**<br>• 0: No internal device port errors are signaled.<br>• 1: The device reported internal fatal/non-fatal errors. |
| 13 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit is set when device experiences a master abort on a transaction it has mastered on IOSF. |
| 12 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit is set when device accesses to IOSF return a failed completion status. |
| 11 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** The IPI interface of the SMT will flag a completer abort for requests to its MMIO space that are greater than 8B in length. |
| 10:9 | 0x0<br>RO | **RSVD:** Reserved |
| 8 | 0x0<br>RW/1C/V | **Master Data Parity Error (MDPD):** This bit is set if the Parity Error Response bit in the PCI Command register set is set and it receives a completion with poisoned data from IOSF. |
| 7 | 0x0<br>RO | **RSVD:** Reserved |
| 6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO | **RSVD:** Reserved |
| 4 | 0x1<br>RO | **Capabilities List (CAPE):** This bit indicates that this device implements a PCI Capability list. |
| 3 | 0x0<br>RO/V | **Interrupt Status (INTS):** When set, this bit indicates that an INTx emulation interrupt is pending internally in this function, it is reset when the internal signal is de-asserted.<br>This bit has meaning only in the legacy interrupt mode.<br>This bit is always 0 when MSI-X has been selected for DMA interrupts. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 49.2.5 RID—Offset 8h

Revision ID Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0.

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:18, F:0] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V/P | **Revision ID (RID):** This value is initially 0 for the first release and is incremented for subsequent releases in order to identify any changes. |

## 49.2.6 CCR—Offset 9h

Class Code Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0. Auto configuration software reads this register to determine the PCI device function.

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:18, F:0] + 9h

**Default:** 00088000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x8<br>RW/O | **Base Class (BASE):** Indicates it is a 'Generic System Peripheral'.<br><br>**Power Well:** PRST |
| 15:8 | 0x80<br>RW/O | **Sub Class (SUB):** Indicates 'Other System Peripheral'.<br><br>**Power Well:** PRST |
| 7:0 | 0x0<br>RO | **RSVD:** Reserved<br><br>**Power Well:** PRST |

## 49.2.7 SMTBAR—Offset 10h

This base address register marks the memory-mapped registers used for SMBus Management Transport functionality. These registers are listed subsequently.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:18, F:0] + 10h

**Default:** 0000000000000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:10 | 0x0<br>RW | **Base Address (MBA):** Base address of SMT control registers. |
| 9:4 | 0x0<br>RO | **Memory Size (MS):** Reserved. |
| 3 | 0x0<br>RO | **Prefetchable (PFMEM):** This bit is hardwired to 0b to indicate non-prefetchable. |
| 2:1 | 0x2<br>RO | **Type (MTYPE):** The SMT MMIO registers should have the ability to be relocated anywhere within 64-bit address space. |
| 0 | 0x0<br>RO | **Memory space indicator (MSI):** Hardwired to 0 to indicate that this is a memory space BAR. |

## 49.2.8 SVID—Offset 2Ch

The Subsystem Vendor ID Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 2Ch

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RW/O | **Subsystem Vendor ID (SVID):** This register uniquely identifies the add-in board or subsystem vendor.<br><br>**Power Well:** PRST |

## 49.2.9 SID—Offset 2Eh

The Subsystem ID Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 2Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/O | **Subsystem ID (SID):** uniquely identifies the add-in board or subsystem.<br><br>**Power Well:** PRST |

## 49.2.10 INTL—Offset 3Ch

Interrupt Line Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0.

This register identifies interrupt line routing information.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:18, F:0] + 3Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Assigned (INTL):** system-assigned value identifies which system interrupt controller's interrupt request has the device's PCI interrupt request routed to it (as specified in the interrupt pin register).<br>A value of FFH signifies 'no connection' or 'unknown'. |

## 49.2.11 INTP—Offset 3Dh

Interrupt Pin Register bit definitions adhere to PCI Local Bus Specification, Revision 3.0. This register must be programmed before legacy interrupts are enabled.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:18, F:0] + 3Dh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RW/L | **Interrupt Pin (INTP):** Indicates which INTx assert/deassert legacy interrupt messages are used.<br>• 01h: Generate INTA.<br>• 02h: Generate INTB.<br>• 03h: Generate INTC.<br>• 04h: Generate INTD.<br>• Others: Reserved.<br><br>**Power Well:** PRST |

## 49.2.12 EXPCAPLST—Offset 40h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 40h

**Default:** 8010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x80<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list ("PMCAPLST—Offset 80h"). |
| 7:0 | 0x10<br>RO | **Capability ID (CAPID):** Identifies the function as PCI Express capable. |

## 49.2.13 DEVCAP—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:18, F:0] + 44h

**Default:** 10008001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved. |
| 28 | 0x1<br>RW/O | **Function Level Reset Capability (FLR):** This field when set indicates this function supports function Level Reset mechanism.<br>Lock key bit for DEVCTL.IFLR.<br>• 0b: Lock.<br>• 1b: Unlocked.<br><br>**Power Well:** PRST |
| 27:26 | 0x0<br>RO | **RSVD:** Reserved |
| 25:18 | 0x0<br>RO | **RSVD:** Reserved |
| 17:16 | 0x0<br>RO | **RSVD1:** Reserved. |
| 15 | 0x1<br>RO | **Role-Based Error Reporting (RBER):** The PCIe cluster supports Role-based Error Reporting. |
| 14:12 | 0x0<br>RO | **RSVD:** Reserved. |
| 11:9 | 0x0<br>RO | **Endpoint L1 Acceptable Latency (EPL1AL):**<br>• 000b Maximum of 1 us<br>• 001b Maximum of 2 us<br>• 010b Maximum of 4 us<br>• 011b Maximum of 8 us<br>• 100b Maximum of 16 us<br>• 101b Maximum of 32 us<br>• 110b Maximum of 64 us<br>• 111b No limit<br>This field does not apply to SMT device. The minimal latency is shown here as default |
| 8:6 | 0x0<br>RO | **Endpoint L0s Acceptable Latency (EPL0AL):**<br>• 000b Maximum of 64 ns<br>• 001b Maximum of 128 ns<br>• 010b Maximum of 256 ns<br>• 011b Maximum of 512 ns<br>• 100b Maximum of 1 us<br>• 101b Maximum of 2 us<br>• 110b Maximum of 4 us<br>• 111b No limit<br>This field does not apply to SMT device. The minimal latency is shown here as default |
| 5 | 0x0<br>RO | **Extended Tag Field Supported (ETFG):**<br>• 0: The device supports 5-bit extended tag<br>• 1: The device supports 8-bit extended tag<br>Currently, only 5-bit tag is supported and validated for the EP |
| 4:3 | 0x0<br>RO | **RSVD:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2:0 | 0x1<br>RO | **Max Payload size Supported (MPSS):** This is an internal device and can send/receive up to 256B packets internally from/to the IOSF fabric. |

## 49.2.14 DEVCTL—Offset 48h

The MPS register field is coded with NFLRST attribute to indicate it is not reset by FLR.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 48h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/V/L | **Initiate Function Level Reset (IFLR):** A write of 1 initiates Function Level Reset.<br>The value read by software from this bit is always 0b.<br>Only applies to end point.<br>This field is locked by the complement of the DEVCAP.FLR LockKey: ~DEVCAP.FLR<br><br>**Power Well:** NFLRST |
| 14:12 | 0x0<br>RO | **Max_Read_Request_Size (MRRS):** This field sets the maximum Read Requests size of the function as a requester on PCIe. |
| 11 | 0x0<br>RW | **Enable No Snoop (ENOSNP):** If this bit this is set, the function is permitted to set the No Snoop bit in the Requester attributes of transactions. |
| 10 | 0x0<br>RO | **RSVD:** Reserved |
| 9 | 0x0<br>RO | **RSVD:** Reserved |
| 8 | 0x0<br>RO | **RSVD:** Reserved |
| 7:5 | 0x0<br>RW | **Maximum Payload Size (MPS):** This field sets maximum TLP payload size for the function.<br>• 000b: 128 bytes maximum payload size (default)<br>• 001b: 256 bytes maximum payload size<br>• Others: Reserved<br><br>**Power Well:** NFLRST |
| 4 | 0x0<br>RO | **RSVD:** Reserved |
| 3 | 0x0<br>RW | **Unsupported Request Reporting Enable (URRE):** This bit controls the enabling of ERR_CORR, ERR_NONFATAL, or ERR_FATAL messages for reporting 'Unsupported Request' errors. |
| 2 | 0x0<br>RW | **Fatal Error Reporting Enable (FERE):** When this bit is set, generation of the ERR_FATAL message is enabled. |
| 1 | 0x0<br>RW | **NonFatal Error Reporting Enable (NFERE):** When this bit is set, generation of the ERR_NONFATAL message is enabled. |
| 0 | 0x0<br>RW | **Correctable Error Reporting Enable (CERE):** When this bit is set, generation of the ERR_CORR message is enabled. |

## 49.2.15 DEVSTS—Offset 4Ah

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 4Ah

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RO/V | **Transactions Pending (TP):** When set, this bit indicates that the device has issued non-posted PCI Express transactions which have not yet completed. |
| 4 | 0x0<br>RO | **RSVD:** Reserved |
| 3 | 0x0<br>RW/1C/V | **Unsupported Request Detected (URD):** This bit indicates that this function received an unsupported request from PCI Express link.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 2 | 0x0<br>RW/1C/V | **Fatal Error Detected (FED):** This bit indicates that the function has detected a Fatal error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 1 | 0x0<br>RW/1C/V | **Non-Fatal Error Detected (NFED):** This bit indicates that the function has detected a Non-Fatal error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |
| 0 | 0x0<br>RW/1C/V | **Correctable Error Detected (CED):** This bit indicates that the function has detected a Correctable error.<br>Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. |

## 49.2.16 PMCAPLST—Offset 80h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 80h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list ("MSICAPLST—Offset 8Ch"). |
| 7:0 | 0x1<br>RO | **Capability ID (CAPID):** Identifies the function as PCI Power Management capable. |

## 49.2.17 MSICAPLST—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NP):** Contains the offset of the next item in the capabilities list. |
| 7:0 | 0x5<br>RO | **Capability ID (CAPID):** Identifies the function as MSI capable. |

## 49.2.18 MSICTL—Offset 8Eh

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + 8Eh

**Default:** 0180h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved. |
| 8 | 0x1<br>RO | **Per-vector masking capable (PVM):** This bit indicates that this device supports MSI per-vector masking. |
| 7 | 0x1<br>RO | **Address 64-Bit Capable (AD64C):** When set, this bit indicates that this function is capable of generating a 64-bit message address.<br>• 1: Function is capable of sending 64-bit message address.<br>• 0: Function is not capable of sending 64-bit message address. |
| 6:4 | 0x0<br>RW | **Multiple Message Enable (MMEN):** These bits are R/W for software compatibility.<br>SMT supports only one message.<br>• 000b = 1 Message.<br>• Others = Reserved. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** SMT supports only one MSI message (power of 2).<br>There is no option of supporting multiple messages. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** When set, MSI is enabled and traditional interrupt pins are not used to generate interrupts.<br>Software must disable INTx and MSI-X for this device when using MSI. |

## 49.2.19 MSIADDR—Offset 90h

**Type:** PCI Configuration Register
(Size: 64 bits)

**Offset:** [B:0, D:18, F:0] + 90h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:2 | 0x0<br>RW | **Address (ADDR):** Message address specified by the system, always Dword aligned. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 49.2.20 MSIDATA—Offset 98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:18, F:0] + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>• 0: Edge Triggered.<br>• 1: Level Triggered. |
| 14 | 0x0<br>RW | **Level (LVL):** If TM is 0h, then this field is a don't care.<br>Edge-triggered messages are always treated as assert messages.<br>If TM is 1h, then:<br>• 0: Deassert Messages<br>• 1: Assert Messages.<br>For level-triggered interrupts, this bit reflects the state of the interrupt input. |
| 13:12 | 0x0<br>RW | **Future:** These bits are don't care for an IOxAPIC interrupt message data field specification. |
| 11:8 | 0x0<br>RW | **Delivery Mode (DM):**<br>• 0000: Fixed: Trigger Mode can be edge or level.<br>• 0001: Lowest Priority: Trigger Mode can be edge or level.<br>• 0010: SMI/PMI/MCA - Not supported via MSI of root port<br>• 0011: Reserved - Not supported via MSI of root port.<br>• 0100: NMI - Not supported via MSI of root port.<br>• 0101: INIT - Not supported via MSI of root port.<br>• 0110: Reserved.<br>• 0111: ExtINT - Not supported via MSI of primary port.<br>• 1000 -1111 - Reserved. |
| 7:0 | 0x0<br>RW | **Interrupt Vector (IV):** The interrupt vector (LSB) will be modified by the IIO to provide context-sensitive interrupt information for different events that require attention from the processor. e.g Hot plug, Power Management and RAS error events. |

## 49.2.21 MSIMSK—Offset 9Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:18, F:0] + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **RSVD0:** Reserved. |
| 0 | 0x0<br>RW | **Mask Bits (MSKB):** For each Mask bit that is set, the PCI Express port is prohibited from sending the associated message.<br>The SMT supports 1 MSI message. Corresponding bits are masked if set to '1'. |

## 49.2.22 PLKCTL—Offset EAh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:18, F:0] + EAh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **RSVD0:** Reserved. |
| 0 | 0x0<br>RW/L | **Capability Lock (CL):** Lock key bit for all RW-L bits (capabilities, next capability pointer, SSID/SVID, slot register, etc) bits for the function.<br>• 1b: Lock.<br>• 0b: Unlocked.<br>This bit is self-locking.<br>Once this bit is set to a 1b, this key bit can not be unlocked. Writing a 0b has no effect on this bit.<br><br>**Power Well:** PRST |

## 49.3 Registers in Memory Space—SMTBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) SMTBAR described in Section 49.2.7, "SMTBAR—Offset 10h" on page 1824.

### 49.3.1 GCTRL—Offset 0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 0h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **RSVD0:** Reserved. |
| 6 | 0x0<br>RW/V/P | **SRST (SoftReset):** Set by the FW to asynchronously Reset the SMT Master and Target logic, Clear all the MMIO registers, and Release the SMBus Clock and Data lines. |
| 5:4 | 0x0<br>RO | **RSVD1:** Reserved. |
| 3 | 0x0<br>RW/V/P | **KILL:** FW sets the bit to cause HW to cause clock low timeout on the SMBus and generate stop after the timeout. |
| 2 | 0x0<br>RW/V/P | **Target Reset (TRST):** Set by FW to force HW to idle its internal SMBus clock and data. |
| 1:0 | 0x0<br>RO | **RSVD2:** Reserved. |

## 49.3.2 SMTICL—Offset 8h

This address register marks the location of where SMT interrupt cause information is written before an MSI is sent. The two least-significant bits are fixed at 00b so that the memory address is Dword-aligned.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [SMTBAR] + 8h

**Default:** 0000000000000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:2 | 0x0<br>RW | **Base Address (BA):** Base address of SMT interrupt cause. |
| 1:0 | 0x0<br>RO | **Memory Size (MS):** Reserved. |

## 49.3.3 ERRINTMSK—Offset 10h

Interrupt masking applies to both MSI and INTx.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 10h

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **RSVD0:** Reserved. |
| 24 | 0x0<br>RW/P | **SMBus Clock Low Timeout Error Interrupt Mask (CKLTO):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.CKLTO being set. |
| 23:18 | 0x0<br>RO | **RSVD1:** Reserved. |
| 17 | 0x0<br>RW/P | **Target Ring Buffer Full Error Interrupt Mask (TRBF):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.TRBF being set. |
| 16 | 0x0<br>RW/P | **Target Ring Buffer Almost Full Error Interrupt Mask (TRBAF):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.TRBAF being set. |
| 15:13 | 0x0<br>RO | **RSVD2:** Reserved. |
| 12 | 0x0<br>RW/P | **IPI Host Interface Error (IHIE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.IHIE being set. |
| 11 | 0x0<br>RW/P | **IPI Master Abort Error (IMAE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.IMAE being set.<br>ERRINTMSK.IMAE: this bit is assumed to be '1' when MSICTL.MSIE is set.<br>IMAE will occur only if PCICMD.BME is clear and MSI requires BME to be set. |
| 10 | 0x0<br>RO | **IPI Transmit Error Interrupt Mask (ITE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.ITE being set.<br>This error condition is deprecated. |
| 9 | 0x0<br>RW/P | **IPI Receive Data Parity Error Interrupt Mask (IRDPE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.IRDPE being set. |
| 8 | 0x0<br>RO | **IPI Receive Error Interrupt Mask (IRE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.IRE being set.<br>This error condition is deprecated. |
| 7:2 | 0x0<br>RO | **RSVD3:** Reserved. |
| 1 | 0x0<br>RW/P | **SPD Write Error Mask (SPDWE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.SPDWE being set. |
| 0 | 0x0<br>RW/P | **CSR Parity Error Interrupt Mask (CPE):** FW sets this bit to mask HW from generating an interrupt due to ERRSTS.CPE being set. |

## 49.3.4 ERRAERMSK—Offset 14h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 14h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **RSVD0:** Reserved. |
| 24 | 0x0<br>RW/P | **SMBus Clock Low Timeout Error AER Mask (CKLTO):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.CKLTO being set. |
| 23:18 | 0x0<br>RO | **RSVD1:** Reserved. |
| 17 | 0x0<br>RW/P | **Target Ring Buffer Full Error AER Mask (TRBF):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.TRBF being set. |
| 16 | 0x0<br>RW/P | **Target Ring Buffer Almost Full Error AER Mask (TRBAF):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.TRBAF being set. |
| 15:13 | 0x0<br>RO | **RSVD2:** Reserved. |
| 12 | 0x0<br>RW/P | **IPI Host Interface Error (IHIE):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.IHIE being set. |
| 11 | 0x0<br>RW/P | **IPI Master Abort Error (IMAE):** FW sets this bit to mask HW from escalating through AER Correctable Internal Error due to ERRSTS.IMAE being set.<br>This error is escalated as Correctable. |
| 10 | 0x0<br>RO | **IPI Transmit Error AER Mask (ITE):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.ITE being set.<br>This error condition is deprecated. |
| 9 | 0x0<br>RW/P | **IPI Receive Data Parity Error AER Mask (IRDPE):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.IRDPE being set. |
| 8 | 0x0<br>RO | **IPI Receive Error AER Mask (IRE):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.IRE being set.<br>This error condition is deprecated. |
| 7:2 | 0x0<br>RO | **RSVD3:** Reserved. |
| 1 | 0x0<br>RW/P | **SPD Write Error AER Mask (SPDWE):** FW sets this bit to mask HW from escalating through AER Correctable Internal Error due to ERRSTS.SPDWE being set. |
| 0 | 0x0<br>RW/P | **CSR Parity Error AER Mask (CPE):** FW sets this bit to mask HW from escalating through AER Uncorrectable Internal Error due to ERRSTS.CPE being set. |

## 49.3.5 ERRSTS—Offset 18h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 18h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **RSVD0:** Reserved. |
| 24 | 0x0<br>RW/1C/V/P | **SMBus Clock Low Timeout Error Status (CKLTO):** This bit is set by HW when SMBus clock is seen low for the time programmed in CLTC.CNT.<br>FW clears this register by writing '1'. |
| 23:18 | 0x0<br>RO | **RSVD1:** Reserved. |
| 17 | 0x0<br>RW/1C/V/P | **Target Ring Buffer Full Error Status (TRBF):** HW sets this bit when it needs to write data and/or header to memory and does not have enough free space in the ring buffer in memory.<br>Once the ring buffer full status bit is set, HW will silently NACK all incoming target transactions on the command phase to indicate that it is busy.<br>FW clears this register by writing '1'. |
| 16 | 0x0<br>RW/1C/V/P | **Target Ring Buffer Almost Full Error Status (TRBAF):** HW sets this bit after it has completed writing data and/or header to memory and checks if there is less than 85B of free space in the ring buffer.<br>FW clears this register by writing '1'. |
| 15:13 | 0x0<br>RO | **RSVD2:** Reserved. |
| 12 | 0x0<br>RW/1C/V/P | **IPI Host Interface Error (IHIE):** HW sets this bit when a read completion returns with non-successful status.<br>FW clears this register by writing '1'. |
| 11 | 0x0<br>RW/1C/V/P | **IPI Master Abort Error (IMAE):** HW sets this bit if a request is made when PCICMD.BME is clear.<br>FW clears this register by writing '1'. |
| 10 | 0x0<br>RO | **IPI Transmit Error Status (ITE):** HW sets this bit if an error is detected in any IPI transmit transaction.<br>This error condition is deprecated. |
| 9 | 0x0<br>RW/1C/V/P | **IPI Receive Data Parity Error Status (IRDPE):** HW sets this bit if a data parity error is detected in any IPI receive transaction.<br>FW clears this register by writing '1'. |
| 8 | 0x0<br>RO | **IPI Receive Error Status (IRE):** HW sets this bit if an error is detected in any IPI receive transaction.<br>This error condition is deprecated. |
| 7:2 | 0x0<br>RO | **RSVD3:** Reserved. |
| 1 | 0x0<br>RW/1C/V/P | **SPD Write Error Status (SPDWE):** HW sets this bit if a write is attempted to SPD address range (A0h-AEh) and write disable (MCTRL.SPDDIS) has been set.<br>FW clears this register by writing '1'. |
| 0 | 0x0<br>RW/1C/V/P | **CSR Parity Error Status (CPE):** HW sets this bit if a parity error is detected in any CSR.<br>FW clears this register by writing '1'. |

## 49.3.6 ERRINFO—Offset 1Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 1Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:13 | 0x0<br>RO/V/P | **Second Error Info (SERRINFO):** Scratchpad. |
| 12:8 | 0x0<br>RO/V/P | **Second Error Pointer (SERRPTR):** Upon the second occurrence of any error, as included in ERRSTS, HW sets this register to point to the specific bit which was set. |
| 7:5 | 0x0<br>RO/V/P | **First Error Info (FERRINFO):** Scratchpad. |
| 4:0 | 0x0<br>RO/V/P | **First Error Pointer (FERRPTR):** Upon the first occurrence of any error, as included in ERRSTS, HW sets this register to point to the specific bit which was set. |

## 49.3.7 MDBA—Offset 100h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [SMTBAR] + 100h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:6 | 0x0<br>RW | **Master Descriptor Base Address (MDBA):** 64Byte aligned base address of the master descriptor ring buffer in FW memory.<br>HW uses this base to add to MSTS.HMTP to calculate the physical address of the memory location it wants to access.<br>FW must update this register only when MCTRL.SS and MSTS.IP bits are cleared. |
| 5:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 49.3.8 MCTRL—Offset 108h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 108h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23:16 | 0x0<br>RW | **Firmware Master Head Pointer (FMHP):** Address offset of the number of descriptors available in the descriptor ring.<br>Updated by FW when it has programmed one or more new descriptors in the master descriptor ring buffer.<br>Always points to the descriptor offset that FW will next write to.<br>FW can increment this register at most to be 'MSTS.HMTP - 1' thus implying that Master Descriptor Ring Buffer can only hold 'MDS - 1' valid descriptors.<br>FW must roll this register to 0 after MDS is reached. |
| 15:5 | 0x0<br>RO | **RSVD1:** Reserved. |
| 4 | 0x0<br>RW | **Master Error Interrupt Enable (MEIE):** Set by FW to enable HW to generate MSI (if MSICTL.MSIE set) if MSTS.MEIS bit is set. |
| 3 | 0x0<br>RW/O/P | **SPD Write Disable (SPDDIS):** This bit must be set to '1' to disable writes to DIMM SPD, which are located in address range A0h-AEh.<br>If set, this address range is unwritable until the next platform reset.<br>Reads are unaffected.<br>This bit is reset only upon platform reset. It should be set by BIOS MRC (Memory Reference Code). |
| 2:1 | 0x0<br>RO | **RSVD2:** Reserved. |
| 0 | 0x0<br>RW/V | **Start/Stop (SS):** Set by the FW when it has programmed at least one descriptor in the descriptor ring.<br>Cleared by FW when it wants HW to stop processing more descriptors.<br>FW must monitor the MSTS.IP bit which indicates that HW is truly stopped.<br>HW clears this register for the following fatal error scenario, regardless of Stop-on-Error (SOE) bit in the master descriptor:<br>• HW receives unsuccessful completion on internal IOSF.<br>HW clears this register for the following non-fatal scenario:<br>• Master descriptor SOE bit was set and the transaction was unsuccessful for any condition as reported in the Status WB Dword. |

## 49.3.9 MSTS—Offset 10Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 10Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23:16 | 0x0<br>RW/V | **Hardware Master Tail Pointer (HMTP):** Descriptor offset in the descriptor ring.<br>Incremented by HW and points to the descriptor that HW is currently processing or will process next.<br>Initialized by FW with the descriptor location where it wants the master to start transaction from (typically 0).<br>This register must only be written by FW when the MCTRL.SS bit is '0' and the MSTS.IP bit is '0'.<br>HW must not increment this pointer in the case that due to the failure of a descriptor-initiated cycle, HW clears the MCTRL.SS bit.<br>HW uses MSTS.HMTP and MCTRL.FMHP to calculate Full and Empty conditions.<br>FW must never initialize this buffer to a value greater than MDS. |
| 15:6 | 0x0<br>RO | **RSVD1:** Reserved. |
| 5 | 0x0<br>RW/1C/V | **Master Interrupt Status (MIS):** Set by HW to indicate that the descriptor that was being processed was successfully done over SMBus.<br>HW proceeds normally to process the next descriptor if the MCTRL.SS bit is set.<br>HW auto-clears this register if INT bit in the descriptor which it finished processing is set and the associated MSI is sent.<br>If INT bit in descriptor is cleared, FW clears this bit by writing '1'. |
| 4 | 0x0<br>RW/1C/V | **Master Error Interrupt Status (MEIS):** Set by HW to indicate that the descriptor that was being processed was not successfully done on SMBus.<br>HW halts the descriptor engine by clearing the MCTRL.SS bit if this bit is set and the SOE bit in the descriptor which was processed was set.<br>HW auto-clears this register if MCTRL.MEIE register is set and the associated Master Error MSI is sent.<br>If MCTRL.MEIE is cleared, FW clears this bit by writing '1'. |
| 3:1 | 0x0<br>RO | **RSVD2:** Reserved. |
| 0 | 0x0<br>RO/V | **In Progress (IP):** Set by the HW to indicate that its DMA processor is active meaning that it is processing a descriptor.<br>HW will clear this bit after it has finished processing the current descriptor (which is the completion of the descriptor's status write back into memory and Interrupt if enabled is sent).<br>This is to aid the stopping of HW even if there are multiple descriptors to be processed. |

## 49.3.10 MDS—Offset 110h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 110h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved. |
| 7:0 | 0x0<br>RW | **Master Descriptor Size (MDS):** Defines the size of the master descriptor ring buffer that is pointed to by MDBA. 0-based number of 16B descriptors.<br>The top of the ring buffer is MDBA+MDS including the last location.<br>A value of '1' represents a total possible capacity of 2 descriptors in the ring, and a value of '255' represents capacity of 256 descriptors in the ring.<br>Used by HW to calculate if it has reached the top of the buffer and must wrap back to 0. |

## 49.3.11 RPOLICY—Offset 114h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00061020h

**BAR and Offset:** [SMTBAR] + 114h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23:16 | 0x6<br>RW | **Time Between Retry Clock Count (TBRCLKCNT):** This register is programmed by the FW to set the retry timer clock count.<br>The reference clock used for this timer count is specified by FW in RPOLICY.TBRBCLK.<br>The combination of the setting in this register and the reference tick used in RPOLICY.TBRBCLK determines how long the HW waits before issuing an automatic retry for master transactions.<br>A value of '0' indicates HW should retry the transaction without any additional delay.<br>***Note:*** Retry count is not applicable for retries due to collision on SMBus. Those must be retried as soon as SMBus is free.<br>This timer must be reloaded every time HW attempts a retry or a completely new cycle is initiated by FW.<br>***Note:*** (FW) Since the timer ticks in register RPOLICY.TBRBCLK are free running and not independently generated by HW, there is a possibility of inaccuracy in delay time. This inaccuracy is at most 1 complete duration of the timer tick, e.g. if 1ms tick is chosen the inaccuracy is never greater than 1ms. Hence FW should use its judgment in which timer tick is selected. Recommendation would be to use a faster tick so that the inaccuracy is smaller. |
| 15:14 | 0x0<br>RO | **RSVD1:** Reserved. |
| 13:11 | 0x2<br>RW | **Collision Retry (COLRTRY):** This field allows the FW to set the number of times master transactions are retried before HW gives up.<br>HW decrements this count after it attempts a master transaction and loses arbitration on SMBus.<br>This timer must be reloaded by HW whenever the RPOLICY.RETRY count is changed (i.e. decremented due to retry or reloaded due to new master cycle initiated).<br>• 000: Reserved.<br>• 001: Retry once (total attempts = 2).<br>• 010: Retry 2 times (total attempts = 3).<br>• 011: Retry 3 times (total attempts = 4).<br>• 100: Retry 4 times (total attempts = 5).<br>• 101: Retry 5 times (total attempts = 6).<br>• 110: Retry 6 times (total attempts = 7).<br>• 111: Retry 7 times (total attempts = 8). |
| 10:8 | 0x0<br>RW | **Time Between Retry Base Clock (TBRBCLK):** This register allows the FW to choose the granularity of the timer tick.<br>• 000: 10us.<br>• 001: 100us.<br>• 010: 1ms.<br>• 011: 10ms.<br>• 100: 100ms.<br>• 101: Reserved.<br>• 110: Reserved.<br>• 111: Reserved.<br>The combination of this register and the RPOLICY.TBRCLKCNT register allows the FW to set mean time between retries on SMBus. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x2<br>RW | **RETRY:** HW will decode this field to determine the number of times to retry before HW gives up.<br>HW must not decrement this count for cycles that are retried because of collision.<br>HW must reload this count every time it starts a new master cycle.<br>HW must not retry cycles if the cycle was unsuccessful due to a clock-low timeout on the SMBus.<br>• 0000: No retries.<br>• 0001: Retry once (total attempts = 2).<br>• 0010: Retry 2 times (total attempts = 3).<br>• 0011: Retry 3 times (Default: total attempts = 4).<br>• 0100: Retry 4 times (total attempts = 5).<br>• 0110: Retry 5 times (total attempts = 6).<br>• 0111: Retry 6 times (total attempts = 7).<br>• ...<br>• 1111: Retry 15 times (Total attempts = 16). |
| 3:0 | 0x0<br>RO | **RSVD2:** Reserved. |

## 49.3.12 TBBA—Offset 200h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [SMTBAR] + 200h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:6 | 0x0<br>RW | **Target Buffer Base Address (TBBA):** 64B aligned base address of the target ring buffer.<br>HW uses this base to add to the HTHP to calculate the physical address of the memory location it wants to access.<br>FW must update this register only when target is disabled and TSTS.IP bit is cleared. |
| 5:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 49.3.13 TCTRL—Offset 208h

For definitions and usage of fields PBC (Payload Byte Count), RBC (Read Byte Count), MTYPE (Message Type), TSTS (Transaction Status), and TTYPE (Transaction Type) in the target header, refer to Figure 20-7, 'Target Header Format' on page 863, Table 20-11, 'Target Header Descriptor' on page 864, Table 20-12, 'Valid Target Descriptor MTYPE and TTYPE Combinations' on page 867, and Table 20-13, 'Target Header Encodings (TSTS) Per Transaction Type (TTYPE)' on page 868.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 208h

**Default:** 00000010h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **RSVD0:** Reserved. |
| 6 | 0x0<br>RW | **Un-Successful Received Target Writes Policy (URxTWP):** FW sets this bit to enable the HW to send data and header for unsuccessful write cycles received by the HW as a target.<br>Since HW is responsible only for writing the payload to memory and sending a descriptor to FW, it is FW responsibility to inspect the transaction data to determine protocol type, payload, etc.<br>The following cycles fall under this policy umbrella:<br>• SMBus Host Notify.<br>• ARP Notify.<br>• ARP Master.<br>• ARP Prepare to ARP.<br>• ARP Assign Address.<br>• ARP General Reset Device command.<br>• ARP Directed Reset Device command to TACTRL.ADDR0 or TACTRL.ADDR1.<br>• SMBus Send Byte command to TACTRL.ADDR0 or TACTRL.ADDR1.<br>• SMBus Write Byte/Word command to TACTRL.ADDR0 or TACTRL.ADDR1.<br>• SMBus Block Write command to TACTRL.ADDR0 or TACTRL.ADDR1 I2C.<br>• Master-TX writes Slave-RX to TACTRL.ADDR0 or TACTRL.ADDR1.<br>The header bits will define the transaction size (PBC and RBC), cycle type (MTYPE and TTYPE), and why the cycle was unsuccessful (TSTS!=0000b).<br>When this bit is cleared, HW only sends data and header to FW of cycles which were successful. |
| 5 | 0x0<br>RW | **Un-Successful Completion Header WB Policy (UCHWBP):** FW sets this bit to enable the HW to send a header for programmed FW cycles that were unsuccessfully completed as a target.<br>The following cycles fall under this policy umbrella:<br>• SMBus Block Reads to GPBR Address (GPBRCTRL.GPTRADR).<br>• SMBus Quick Command to TACTRL.ADDR0 or TACTRL.ADDR1.<br>• ARP Directed Get UDID command to TACTRL.ADDR0 or TACTRL.ADDR1.<br>• ARP General Get UDID command.<br>The header bits will define the transaction size (PBC and RBC), cycle type (MTYPE and TTYPE), and why the cycle was unsuccessful (TSTS!=0000b).<br>When this bit is cleared, no header writeback is done by HW.<br>If General Get UDID command is received and that UDID Address Resolved (AR) flag is set, HW will NACK the ARP 'command' byte to indicate it has a valid assigned slave address (its ARP is complete), and it should not send any unsuccessful completion header to FW. |
| 4 | 0x1<br>RW | **Successful Completion Header WB Policy (SCHWBP):** FW sets this bit to enable the HW to send a header for programmed FW cycles that were successfully completed as target on SMBus.<br>The following cycles fall under this policy umbrella:<br>• SMBus Block Reads to GPBR Address (GPBRCTRL.GPTRADR).<br>• Cycles to ADDR0 or ADDR1 (TACTRL.ADDR0 or TACTRL.ADDR1).<br>• Cycles to Default Address (C2h).<br>• Cycles to SMBus or ARP Host (10h).<br>The header bits will define the transaction size (PBC and RBC), cycle type (MTYPE and TTYPE), and successful completion status (TSTS=0000b).<br>When this bit is cleared, no header writeback is done by HW. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:1 | 0x0<br>RO | **RSVD1:** Reserved. |
| 0 | 0x0<br>RW | **Target Interrupt Enable (TIE):** FW sets this bit to enable HW to send MSI when it writes header to memory, i.e. when TSTS.TIS is set. |

## 49.3.14 TSTS—Offset 20Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 20Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved. |
| 11:8 | 0x0<br>RO/V | **Target Cycle In Progress (TCIP):** This field allows FW to check which logical target address HW is currently servicing as a target.<br>This field is set by HW at the same time as TSTS.IP transitions from 0 to 1.<br>• 0000: Default SMBus transaction to C2h.<br>• 0001: Transaction targeting address (TACTRL.ADDR0) of UDID0.<br>• 0010: Transaction targeting address (TACTRL.ADDR1) of UDID1.<br>• 0100: Host Notify transaction targeting 10h.<br>• 0101: Generic Programmable Block Read to programmed address in GPBRCTRL.GPTRADR.<br>• Others: Reserved.<br>This field is valid only when the TSTS.IP bit is set.<br>This field is consistent with Target Header Descriptor TTYPE field. |
| 7:2 | 0x0<br>RO | **RSVD1:** Reserved. |
| 1 | 0x0<br>RO/V | **In Progress (IP):** This bit is set by HW when it is processing a target cycle received on SMBus. This is the phase after the HW has ACKed a matching address until Stop is asserted by the external initiator, target buffer is emptied, and interrupt if enabled is sent to FW.<br>This bit is cleared otherwise. |
| 0 | 0x0<br>RW/1C/V | **Target Interrupt Status (TIS):** This bit is set by HW after it has written status header in target ring buffer as an indication of interrupt generation.<br>• The conditions which result in a status header being written are governed by these policies: TCTRL.SCHWBP: upon successful completion.<br>• TCTRL.UCHWBP: upon unsuccessful completion.<br>• TCTRL.URxTWP: upon unsuccessful completion of writes to target.<br>If TCTRL.TIE is cleared, FW must write '1' to clear this bit. It TCTRL.TIE register is set and global MSICTL.MSIE is set, HW sends the MSI and auto-clears this bit. |

## 49.3.15 TBS—Offset 210h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 210h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:2 | 0x0<br>RW | **Target Buffer Size (TBS):** Defines the size in bytes of the target ring buffer in memory which must be Dword aligned.<br>Bits[1:0] are reserved such that FW always provides HW a buffer in granularity of 4Bytes.<br>• 0x0000: 4B.<br>• 0x0004: 8B.<br>• ...<br>• 0xFFFC: 65536B (~64KB). |
| 1:0 | 0x0<br>RO | **RSVD1:** Reserved. |

## 49.3.16 HTHP—Offset 218h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 218h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:2 | 0x0<br>RW/V | **Hardware Target Buffer Head Pointer (HTBHP):** Dword offset from the base address within the target ring buffer.<br>Updated by HW to indicate the offset within the ring buffer where the HW is writing payload data/ header or will be writing to next.<br>HW uses HTHP and FTTP and TBS to calculate buffer full, empty, and wrap conditions.<br>FW must never initialize this to a value greater than TBS.TBS or undefined HW behavior may occur. |
| 1:0 | 0x0<br>RO | **RSVD1:** Reserved. |

## 49.3.17 FTTP—Offset 21Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 21Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:2 | 0x0<br>RW | **Firmware Target Buffer Tail Pointer (FTBTP):** Dword offset from the base address within the target ring buffer.<br>Updated by FW and points to the offset FW is reading from or will read from next.<br>FW must never increment this register beyond the value of the HTHP.HTBHP register. |
| 1:0 | 0x0<br>RO | **RSVD1:** Reserved. |

## 49.3.18 TRxCTRL—Offset 220h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000058h

**BAR and Offset:** [SMTBAR] + 220h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Descriptio |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved. |
| 7:0 | 0x58<br>RW | **Max Receive Bytes (MRxB):** This register sets the limit for the maximum number of bytes (address included) that SMT can receive as a target of write transactions.<br>Once this limit is reached, HW must NACK the following received bytes.<br>The encoding is 0's based, i.e. '0' means 1 byte (address only) and '255' means 256 bytes.<br>• 00h: Reserved.<br>• 01h: 2 bytes (address + 1 additional byte).<br>• ...<br>• EFh: 240 bytes (address + 239 additional bytes).<br>• F0h: Reserved.<br>• ...<br>• FFh: Reserved.<br>HW must always track the received number of bytes for I2C transactions to ensure that it never overflows the count programmed in this register.<br>When this count is met, HW must NACK any following received bytes.<br>(It is expected that bounded protocols, like SMBus which announces a byte count, provide protocol-level control to ensure write count is not exceeded. Should HW NACK such an overflowing transaction it is the responsibility of the bus master to recover and retry.)<br>Default is defined for MCTP with 80B MCTP packet payload:<br>(8B) MCTP SMBus Packet Header (includes 4B SMBus medium-specific header and 4B MCTP transport header)<br>(80B) MCTP Packet Payload (1B) SMBus PEC = 89B --) 88d (0's-based encoding). |

## 49.3.19 TRxSTS—Offset 224h

This register is updated by HW for cycles received as a target. HW updates this register with every new target cycle received even during EP well power available. The contents are similar to the target header updates by HW, so refer to the header fields for explanation, unless noted below.

This register is primarily a debug feature. This register is updated based on HW responding to transactions on SMBus. There is no ordering maintained w.r.t. updating this register and header/data write back to memory.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 224h

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **Payload Byte Count (PBC):** HW updates this field to indicate how many bytes were ACKed on SMBus sent by the external master and stored within the target buffer locally.<br>For writes coming in, this data will include the sum total of all bytes ACKed from 'start' to 'stop'.<br>For error scenarios where the master drives more data than expected or command code does not match, HW will update the count with all bytes ACKed on SMBus from 'Start' until the first byte NACKed by HW (this byte is not added to the count). |
| 23:22 | 0x0<br>RO | **RSVD0:** Reserved. |
| 21:16 | 0x0<br>RO/V | **Read Byte Count (RBC):** This byte count is for externally generated read transactions only.<br>HW updates this field to indicate the sum total of data bytes it successfully provided over SMBus to the external master. |
| 15:12 | 0x0<br>RO | **RSVD1:** Reserved. |
| 11:8 | 0x0<br>RO/V | **MTYPE:** Check the Target header description. |
| 7:4 | 0x0<br>RO/V | **TSTS:** Check the Target header description. |
| 3:0 | 0x0<br>RO/V | **TTYPE:** Check the Target header description. |

## 49.3.20 TACTRL—Offset 228h

Each address supported by SMT device must be unique since they all logically reside on the same SMBus segment.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 228h

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:9 | 0x0<br>RW | **Address1 (ADDR1):** FW programs this field with an address of the SMT logic. |
| 8 | 0x0<br>RO | **RSVD1:** Reserved. |
| 7:1 | 0x0<br>RW | **Address0 (ADDR0):** FW programs this field with an address of the SMT logic. |
| 0 | 0x0<br>RO | **RSVD2:** Reserved. |

## 49.3.21 TPOLICY—Offset 22Ch

The target addresses referenced above are located in the following registers:

- TACTRL.ADDR0
- TACTRL.ADDR1
- GPBRCTRL.GPTRADR

FW can write to this register to enable/disable functionality pertaining to a target address. FW requests HW to update the new policy by setting TPOLICY.PTUREQ register.

HW waits until all the pending target cycles are over, and copies the contents of this register into internal non-FW-visible register bits. Once it is done, HW clears the TPOLICY.PTUREQ register as confirmation to FW that the new policy is in effect. FW can always read this register and see the current policy that HW is using if the PTUREQ bit is cleared. Any setting or clearing of the policies in this register without setting the PTUREQ bit first must not cause HW to change policies.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 22Ch

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/V/P | **Target Policy Update Request (PTUREQ):** FW sets this bit to indicate to the HW that it wants to update registers/policies in the target interface.<br>The new policy that the FW needs is programmed in the lower word of this register.<br>HW clears this bit when it has updated the new policy.<br>When this bit is cleared the other registers indicate the current policy in effect in HW. |
| 30:11 | 0x0<br>RO | **RSVD0:** Reserved. |
| 10 | 0x0<br>RW | **C2_BSY:** When set forces HW to ACK only matching C2h address byte and NACK in any subsequent bytes to indicate device is busy. |
| 9 | 0x0<br>RW | **Host Busy (HOSTBSY):** When set forces HW to ACK only matching 10h address byte and NACK in any subsequent bytes to indicate device is busy. |
| 8 | 0x0<br>RW | **ADDR1 Busy (ADDR1BSY):** When set forces HW to ACK only matching ADDR1 address byte and NACK in any subsequent bytes to indicate device is busy. |
| 7 | 0x0<br>RO | **RSVD1:** Reserved. |
| 6 | 0x0<br>RW | **ADDR0 Busy (ADDR0BSY):** When set forces HW to ACK only matching ADDR0 address byte and NACK in any subsequent bytes to indicate device is busy. |
| 5 | 0x0<br>RO | **RSVD2:** Reserved. |
| 4 | 0x0<br>RW/V/P | **Device Address0 Enable (ADDR0_EN):** When set, enables target logic to ACK ADDR0 address on the SMBus. Fuse load. |
| 3 | 0x0<br>RW/V/P | **Host SMBus Address Enable (Host_SMBADDR_EN):** When set, enables HW to accept write cycles targeting address 10h.<br>This bit must be set to '1' when SoC is the SMBus Segment Owner. Fuse load. |
| 2 | 0x0<br>RW | **Default SMBus Address Enable (DEF_SMBADDR_EN):** When set, enables HW to ACK cycles targeting address C2h, which is the SMBus default address. |
| 1 | 0x0<br>RW/V/P | **Device Address1 Enable (ADDR1_EN):** When set, enables target logic to ACK ADDR1 address on the SMBus. Fuse load. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RW | **Target Enable (TGTEN):** Set to '1' by the FW after it has programmed the base address in the Target Data Ring Buffer and updated target policies.<br>While in default state '0' all bytes will be NACKed on SMBus including the address byte of the enabled addresses.<br>TGTEN applies to all target addresses: ADDR0_EN, ADDR1_EN, Host_SMBADDR_EN, DEF_SMBADDR_EN, and GPBRCTRL.EN. |

## 49.3.22 GPBRCTRL—Offset 240h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 240h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/V/P | **Enable (EN):** This bit is set by FW to enable HW to respond to a block read command to address GPBRCTRL.GPTRADR and command GPBRCTRL.CMD.<br>HW clears this bit after HW ACKs the command code (CMD) on the Block Read command if GPBRCTRL.HWCLRDIS is cleared. |
| 30 | 0x0<br>RW | **PEC Enable (PECEN):** This bit is set by FW to indicate that HW must provide PEC at the end of the transaction to the external master. |
| 29 | 0x0<br>RW | **HW Clear Disable (HWCLRDIS):** This bit is set by FW to disable the HW from clearing of the GPBRCTRL.EN bit.<br>This functionality allows HW to continuously provide current data in the GPRDx registers to the external master without FW involvement.<br>When this bit is set, HW must start sending data from byte 0 (of the buffer behind GPBRDBUF) when responding to a new Block Read cycle. |
| 28 | 0x0<br>RW/V | **GPBR Buffer Contents Reset (BUFRST):** This bit is set by FW to reset (zero) the contents of the GPBR buffer ("GPBRDBUF—Offset 244h").<br>HW clears this bit after it has reset the GPBR buffer contents. |
| 27 | 0x0<br>RW/V | **GPBR Buffer Pointer Reset (PTRRST):** This bit is set by FW to reset the pointer into the GPBR buffer ("GPBRDBUF—Offset 244h").<br>HW clears this bit after it has reset the GPBR buffer pointer. |
| 26:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23:16 | 0x0<br>RW | **Byte Count (BC):** Byte count indicating how many bytes are programmed in the Generic Read Data buffer ("GPBRDBUF—Offset 244h") by FW.<br>The maximum count is 32B, which is the maximum number of bytes that the buffer can hold.<br>On the block read, after HW matches the address and command, HW sends the contents of this byte on the SMBus and then sends out those many bytes from the Generic Read Data Buffer starting from the LSB of GPBRDBUF0 register, i.e. the byte 0 of GPBRDBUF0. |
| 15:8 | 0x0<br>RW | **Command (CMD):** Command byte of the block read command. |
| 7:0 | 0x0<br>RW/V/P | **General Purpose Target Read Address (GPTRADR):** Target address of the externally initiated Block Read command cycle. Bit [0] of this register is the 'Enable' for this address.<br>HW does not take any action based on the value of bit [0].<br>It is simply provided to FW for reading. Fuse load. |

## 49.3.23 GPBRDBUF—Offset 244h

This register is used to load data into the generic programmable read data buffer. The buffer is used to provide a mechanism of supporting generic programmable reads (GPBR) by an external SMBus master under FW initiated flow. The data buffer is filled by FW initiating DW writes to this register. A hidden pointer always points to the next available buffer location reading and writing to GPBRDBUF cause the pointer to increment automatically.

Because the pointer auto-increments after each write, a read of GPBRDBUF will NOT return the most recently written data. Normal usage requires the buffer contents to be cleared prior to filling it with a new data payload, therefore a read (of the next 'empty' location) will return 0s (See also GPBRCTRL.BUFRST and GPBRCTRL.PTRRST).

HW provides read data payload starting from bit [7:0] of Dword0 of the buffer, then bit [15:8] of Dword0, and so on. All bytes must be programmed in the sequence to be sent out without gaps. HW will simply count the number of bytes indicated in the GPBRCTRL.BC register and start sending them out sequentially starting at byte 0 of Dword 0 until the byte count is decremented to 0.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [SMTBAR] + 244h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Data DWORD (GPBRD):** Reserved. |

## 49.3.24 SMTARPCTRL—Offset 280h

The flags AV0 and AR0 refer to UDID0 and its corresponding target address TACTRL.ADDR0. The flags AV1 and AR1 refer to UDID1 and its corresponding target address TACTRL.ADDR1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 280h

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Disable 'Notify ARP Master' (NOTIFYD):** If asserted, the SMT will not issue a 'Notify ARP Master' command.<br>Otherwise, if SMT is not the ARP Host, it is permitted to issue a 'Notify ARP Master' command. |
| 30:7 | 0x0<br>RO | **RSVD0:** Reserved. |
| 6 | 0x0<br>RW | **Address Resolved (AR1):** If asserted, the device's slave address has been resolved by the ARP Master. |
| 5 | 0x0<br>RW | **Address Valid (AV1):** If asserted, the device's slave address is valid. |
| 4 | 0x0<br>RW | **UDID1 Enable (UDID1E):** If cleared the SMT will not ACK or respond to ARP transactions directed to UDID1. |
| 3 | 0x0<br>RO | **RSVD1:** Reserved |
| 2 | 0x0<br>RW | **Address Resolved (AR0):** If asserted, the device's slave address has been resolved by the ARP Master. |
| 1 | 0x0<br>RW | **Address Valid (AV0):** If asserted, the device's slave address is valid. |
| 0 | 0x0<br>RW | **UDID0 Enable (UDID0E):** If cleared the SMT will not ACK or respond to ARP transactions directed to UDID0. |

## 49.3.25 UDID0—Offset 290h

These bits are programmed by FW and are provided by the HW to external master when it initiates a Directed Get UDID Block Read command (with PEC) to UDID0's SMBus target address.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [SMTBAR] + 290h

**Default:** 8086000000000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:48 | 0x8086<br>RO | **Subsystem Vendor ID (SVID):** Subsystem Vendor ID Indicates that the device subsystem vendor is Intel. |
| 47:32 | 0x0<br>RO | **Subsystem Device ID (SDID):** Subsystem Device ID. |
| 31:0 | 0x0<br>RW/V/P | **Vendor Specific ID (VSID):** Vendor Specific ID must uniquely identify this device when dynamic, volatile addressing is enabled.<br>At least 24 bits should be provided from fuses to uniquely identify over 16 million the SoC devices (without resorting to other revision fields). Fuse load |

## 49.3.26 UUDID0—Offset 298h

These bits are programmed by FW and are provided by the HW to external master when it initiates a Directed Get UDID Block Read command (with PEC) to UDID0 SMBus target address.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [SMTBAR] + 298h

**Default:** 8108808600000024h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x81<br>RW | **Device Capabilities (DEVCAP):** Device Capabilities by default indicate that the device is a dynamic and volatile address device which supports PEC.<br>FW may modify the capability, for instance to behave as a fixed-address device. |
| 55:48 | 0x8<br>RO | **Version/Revision (RID):** Version/ Revision indicating UDID version 1 and Revision 0 of the silicon. |
| 47:32 | 0x8086<br>RO | **Vendor ID (VID):** Vendor ID indicates that the device vendor is Intel. |
| 31:16 | 0x0<br>RO | **Device ID (DID):** Device ID. |
| 15:0 | 0x24<br>RW | **Interface (INTF):** Interface field indicates that the device supports SMBus 2.0 and its ASF capabilities bit set. |

## 49.3.27 UDID1—Offset 2A0h

These bits are programmed by FW and are provided by the HW to external master when it initiates a Directed Get UDID Block Read command (with PEC) to UDID1 SMBus target address.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [SMTBAR] + 2A0h

**Default:** 8086000000000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:48 | 0x8086<br>RO | **Subsystem Vendor ID (SVID):** Subsystem Vendor ID Indicates that the device subsystem vendor is Intel. |
| 47:32 | 0x0<br>RO | **Subsystem Device ID (SDID):** Subsystem Device ID. |
| 31:0 | 0x0<br>RW/V/P | **Vendor Specific ID (VSID):** Vendor Specific ID must uniquely identify this device when dynamic, volatile addressing is enabled.<br>At least 24 bits should be provided from fuses to uniquely identify over 16 million the SoC devices (without resorting to other revision fields). Fuse load |

## 49.3.28 UUDID1—Offset 2A8h

These bits are programmed by FW and are provided by the HW to external master when it initiates a Directed Get UDID Block Read command (with PEC) to UDID1's SMBus target address.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**BAR and Offset:** [SMTBAR] + 2A8h

**Default:** 8108808600000024h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x81<br>RW | **Device Capabilities (DEVCAP):** Device Capabilities by default indicate that the device is a dynamic and volatile address device which supports PEC.<br>FW may modify the capability, for instance to behave as a fixed-address device. |
| 55:48 | 0x8<br>RO | **Version/Revision (RID):** Version/ Revision indicating UDID version 1 and Revision 0 of the silicon. |
| 47:32 | 0x8086<br>RO | **Vendor ID (VID):** Vendor ID indicates that the device vendor is Intel. |
| 31:16 | 0x0<br>RO | **Device ID (DID):** Device ID. |
| 15:0 | 0x24<br>RW | **Interface (INTF):** Interface field indicates that the device supports SMBus 2.0 and its ASF capabilities bit set. |

## 49.3.29 SPGT—Offset 300h

Setting of these timing parameters by FW, will affect both master and target timing on the physical.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 300h

**Default:** 00000005h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RW/V/P | **SMBus Speed (SPD):** Value to indicate what speed physical bus must operate.<br>It is up to the system designer to ensure that all devices on the same SMBus segment are capable of operating at the programmed speed.<br>• 00: Standard (80 kHz).<br>• 01: Standard (100 kHz).<br>• 10: Fast Mode (400 kHz).<br>• 11: Fast Mode Plus (1 MHz). |
| 29 | 0x0<br>RW | **Speed Lock (SPDLK):** FW sets this register if it overrides the settings of SPGT.SPD register.<br>If this bit is set, HW must not load the SPGT.SPD from the soft strap. |
| 28:20 | 0x0<br>RO | **RSVD0:** Reserved. |
| 19:16 | 0x0<br>RW | **T-Hold Data (THDDAT):** The value in this field will be added to the Thddat timing parameter as defined in the SMBus 2.0 spec.<br>• 0000: 0 Clocks (0ns).<br>• 0001: 1 Clocks (10ns).<br>• 0010: 2 Clocks (20ns).<br>• ...<br>• 1111: 15 Clocks (150ns). |
| 15:12 | 0x0<br>RO | **RSVD1:** Reserved. |
| 11:8 | 0x0<br>RW | **T-Setup Data (TSUDAT):** The value in this field will be added to the Tsudat timing parameter as defined in the SMBus 2.0 spec.<br>• 0000: 0 Clocks (0ns).<br>• 0001: 1 Clocks (10ns).<br>• 0010: 2 Clocks (20ns).<br>• ...<br>• 1111: 15 Clocks (150ns). |
| 7:0 | 0x5<br>RW | **De-Glitch (DG):** The value in this field defines the minimum de-glitch pulse that the de-glitch logic will reject from the SMBus clock and data lines.<br>• 000000xx: No de-glitch.<br>• 00000100: 4 Clocks (40ns glitch pulse).<br>• 00000101: 5 Clocks (50ns glitch pulse).<br>• ...<br>• 11111111: 255 Clocks (2550ns glitch pulse). |

## 49.3.30 SPMT—Offset 304h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 08080000h

**BAR and Offset:** [SMTBAR] + 304h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x8<br>RW | **Time high (Thigh):** The value in this field will be added/subtracted to the Thigh timing parameter as defined in the SMBus 2.0 spec.<br>• 00000000: 0 Clock (0ns).<br>• 00000001: -1 Clock (-10ns).<br>• ...<br>• 00000111: -7 Clocks (-70ns).<br>• 00001000: 0 Clock (0ns).<br>• 00001001: +1 Clock (+10ns).<br>• ...<br>• 11111111: +247 Clocks (+2,470ns).<br>These are offsets that control the range of the nominal Thigh value in the PHY layers. |
| 23:16 | 0x8<br>RW | **Time low (Tlow):** The value in this field will be added/subtracted to the Tlow timing parameter as defined in SMBus 2.0 spec. See Thigh field above for encoding. |
| 15:12 | 0x0<br>RW | **Thdsta:** The time value in this field will be added/subtracted to the Thdsta timing parameter as defined in SMBus 2.0 spec.<br>• 1111: -7 Clocks (-70ns).<br>• 1110: -6 Clocks (-60ns).<br>• ...<br>• x000: 0 Clock (0ns).<br>• 0001: +1 Clock (+10ns).<br>• ...<br>• 0111: +7 Clocks (+70ns). |
| 11:8 | 0x0<br>RW | **Tsusta:** The time value in this field will be added/subtracted to the Tsusta timing parameter as defined in SMBus 2.0 spec. See Thdsta field above. |
| 7:4 | 0x0<br>RW | **Tbuf:** The time value in this field will be added/subtracted to the Tbuf timing parameter as defined in SMBus 2.0 spec. See Thdsta field above. |
| 3:0 | 0x0<br>RW | **Tsusto:** The time value in this field will be added/subtracted to the Tsusto timing parameter as defined in SMBus 2.0 spec. See Thdsta field above. |

## 49.3.31 SPST—Offset 308h

These values should only be changed when the physical is idle. These timings are based off the 100-MHz PHY clock. A higher-frequency PHY clock will scale the timings accordingly.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 308h

**Default:** 00000000h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **Slave Tlow Extension (Slave_Tlow_Ext):** The time value in this field will be added to the SMBCLK Tlow extension timing parameter on a per bit basis.<br>• 00000000: 0 Clock (0ns).<br>• 00000001: 1 Clock (10ns).<br>• ----<br>• 11111111: 255 Clocks (2,550ns).<br>These Tlow extensions are only visible when set ) Tlow:min as defined in the SMBus 2.0 spec.<br>The slave will attempt to stretch the CLK on every bit, thus slowing down the entire bus segment, regardless of whether it is currently the target addressed by an external master. |
| 23:12 | 0x0<br>RO | **RSVD0:** Reserved. |
| 11:8 | 0x0<br>RW | **Slave Tsusta (Slave_Tsusta):** The time value in this field will be added/subtracted to the detection of Tsusta timing parameter as defined in SMBus 2.0 spec.<br>• 1111: -7 Clocks (-70ns).<br>• 1110: -6 Clocks (-60ns).<br>• ----<br>• x000: 0 Clock (0ns).<br>• 0001: +1 Clock (+10ns).<br>• ----<br>• 0111: +7 Clocks (+70ns).<br>These are offsets that control the range of the nominal Tsusta value detected as a slave the PHY layers. |
| 7:4 | 0x0<br>RW | **Slave Tthdsta (Slave_Tthdsta):** The time value in this field will be added/subtracted to the detection of Thdsta timing parameter as defined in SMBus 2.0 spec. See Slave_Tsusta field above. |
| 3:0 | 0x0<br>RW | **Slave Ttsusto (Slave_Ttsusto):** The time value in this field will be added/subtracted to the detection of Tsusto timing parameter as defined in SMBus 2.0 spec. See Slave_Tsusta field above. |

## 49.3.32 SMBFT—Offset 30Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00002024h

**BAR and Offset:** [SMTBAR] + 30Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Fair Flag (FF):** HW sets this bit when it has mastered a cycle by processing a descriptor in which 'FAIR' control bit was set.<br>This flag when set is an internal state for the HW to wait longer before it attempts to master the next cycle on SMBus.<br>The next cycle does not necessarily have to be descriptor-based or have the 'FAIR' bit set in the descriptor if it is descriptor-based.<br>Mastering a cycle is defined as when HW attempts to do a SMBus block write and in the data phase following byte count it sees its own SMBus address (with RW bit = 1) on the SMBus.<br>This is per MCTP spec. HW clears this flag after the expiry of SMBFT.SFIC. |
| 30:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:8 | 0x20<br>RW | **SMBus Start Delay (SSD):** This count is the number of ticks (in increments of 1us) that HW must count after SMBFT.FF is cleared before it can attempt to master another SMBus cycle.<br>HW must look for transition of SMBFT.FF bit to transition from 1-)0 before it starts counting off this delay time.<br>• 0h: Reserved (see note).<br>• 1h: Reserved (see note).<br>• 2h: 2us.<br>• ...<br>• FFh: 255us.<br>***Note:*** The base delay for this in each timing mode is defined as T_IDLE_DELAY in MCTP spec.<br>For Standard Mode (100kHz) or lower SMBus operation the minimum is 31us.<br>For Fast Mode (400kHz) the minimum is 16us.<br>For Fast Mode Plus (1MHz) the minimum is 10us (MCTP over SMBus/I2C spec has no definition of timings for Fast Mode Plus).<br>The 1us tick used by the HW to count off this delay is not internally generated in SMBus controller, hence the count could be off by a 1us tick therefore 1us guard band should be added to compensate for this. |
| 7:0 | 0x24<br>RW | **SMBus Fair Idle Count (SFIC):** This count represents the minimum SMBus Idle Time (in increments of 1us) that the HW must detect after SMBus Stop is seen before clearing the SMBFT.FF register.<br>• 0h: reserved (see note).<br>• 1h: reserved (see note).<br>• 2h: 2us.<br>• ...<br>• FFh: 255us.<br>***Note:*** The base delay for Fair Idle Count timings is designed to match the T_IDLE_WINDOW minimum plus T_BUF.<br>For Standard Mode (100kHz) or lower SMBus operation the window is 30-60us to which must be added 4.7us (T_BUF).<br>For Fast Mode (400kHz) the window is 5-20us to which must be added 1.3us (T_BUF).<br>For Fast Mode Plus (1MHz) the window is 3us to which must be added 0.5us (T_BUF) (MCTP over SMBus/I2C spec has no definition of timings for Fast Mode Plus.).<br>The 1us tick used by the HW to count off this delay is not internally generated in SMBus controller, hence the count could be off by a 1us tick therefore 1us guard band should be added to compensate for this. |

## 49.3.33 CLTC—Offset 310h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000015E1h

**BAR and Offset:** [SMTBAR] + 310h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:4 | 0x15e<br>RW | **Count (CNT):** FW sets this to the desired count that the HW must count while SMBus clock is low before flagging SMBus Clock Low Timeout.<br>• 0h: Reserved (see note for BCLK).<br>• 1h: Reserved (see note for BCLK).<br>• Others: Count per BCLK.<br>Default value is set to provide 35ms clock low timeout detection (for default 100us BCLK). |
| 3 | 0x0<br>RO | **RSVD1:** Reserved. |
| 2:0 | 0x1<br>RW | **Base Clock (BCLK):** FW programs this register to set the granularity of the count.<br>The value in this register selects which timer ticks are used to count off the value programmed in CLTC.CNT register.<br>• 000: 10us.<br>• 001: 100us (default).<br>• 010: 1ms.<br>• 011: 10ms.<br>• 100: 100ms.<br>• Others: Reserved.<br>The timer ticks are not internally generated by HW hence the count could be off by 1 timer tick. |

## 49.3.34 DCLKGT—Offset 380h

By default, clock gating is enabled in HW. However in case clock gating has to be disabled in HW, FW can set these register bits to disable clock gating at the individual levels defined below.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 80000000h

**BAR and Offset:** [SMTBAR] + 380h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Global Clock Gating Enable (GCGE):** This bit when set enables dynamic clock gating of the transaction, link, and PHY layers.<br>When set, HW enables global clock gating and the other register policies are followed (DCLKGT bits [6:0]). When cleared, HW disables all clock gating and overrides the policies set by DCLKGT bits [6:0]. |
| 30:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 49.3.35 SUSCHKB—Offset 384h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000011h

**BAR and Offset:** [SMTBAR] + 384h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD0:** Reserved. |
| 5 | 0x0<br>RW | **NOA Select (NOASEL):** When set allows NOA signals of instance #2 of SMT to be viewed on NOA for debug purposes. |
| 4 | 0x1<br>RW | **Auto Slave Clock Low Disable (ASCKLDIS):** When set disables the auto-clock low stretching by the slave. |
| 3 | 0x0<br>RW | **25ms Timeout Detection Enable (TODE25):** When set bypasses the FW controlled Clock low timeout detection and reverts back to fixed 25ms Clock low timeout detection. |
| 2 | 0x0<br>RO | **RSVD1:** Reserved. |
| 1 | 0x0<br>RW | **Ignore R/W bit on Start Address (IRWST):** When set, HW ignores the value of R/W bit on the start address for all of its SMBus target addresses.<br>HW just matches bits [7:1] to ACK the address.<br>If this bit is clear and R/W bit does not match, HW will ACK the address on SMBus, but will NACK all further bytes until stop is seen.<br>There is no FW notification for this transaction.<br>Cycles targeting address C2h and 10h are also applicable to this policy. |
| 0 | 0x1<br>RW | **Ignore R/W bit on Repeated Start Address (IRWRST):** When set, HW ignores the value of R/W bit on the repeated start address for an SMBus transaction.<br>HW just matches bits [7:1] on repeated start address.<br>Cycles targeting address C2h are also applicable to this policy. |

## 49.3.36 DBCTRL—Offset 388h

*Note:* For FW: HW provides the FW a generic ability to bit bang on the SMBUs. It is however the responsibility of the FW that: HW DMA engine is completely stopped and no transactions are pending within HW, HW target interface is disabled, and SMBus is idle prior to bit banging. FW must guarantee all SMBus protocol and timing.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [SMTBAR] + 388h

**Default:** 00000003h

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Enable (EN):** This register is set by FW when it is bit banging on SMBus.<br>This bit must always be set when FW is bit banging on SMBus.<br>FW must clear this bit when it is done bit banging.<br>It is recommended that FW first set the enable bit before it starts bit banging on the clock and data line. |
| 30:2 | 0x0<br>RO | **RSVD0:** Reserved. |
| 1 | 0x1<br>RW | **SMBus Clock (SMBCK):** This register provides ability for the FW to bit bang directly on the SMBus Clock pin.<br>When FW writes a '0' to this register, HW will drive the SMBus Clock low. |
| 0 | 0x1<br>RW | **SMBus Data (SMBDT):** This register provides ability for the FW to bit bang directly on the SMBus Data pin.<br>When FW writes a '0' to this register, HW will drive the SMBus Data low. |

## 49.3.37 DBSTS—Offset 38Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000300h

**BAR and Offset:** [SMTBAR] + 38Ch

**SMTBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMTBAR Reference:** [B:0, D:18, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **RSVD0:** Reserved. |
| 9 | 0x1<br>RO/V | **SMBus Clock (SMBCK):** This register shows the live state of SMBus clock signal as seen by HW from the buffer after HW deglitch and synchronization. |
| 8 | 0x1<br>RO/V | **SMBus Data (SMBDT):** This register shows the live state of SMBus data signal as seen by HW from the buffer after HW deglitch and synchronization. |
| 7 | 0x0<br>RO/C | **SMBus Clock Change (SMBCKC):** This register when set indicates that there was a 1-)0 or 0-)1 transition on SMBus clock line.<br>This register is cleared by HW on reads by FW. |
| 6 | 0x0<br>RO/C | **SMBus Data Change (SMBDC):** This register when set indicates that there was a 1-)0 or 0-)1 transition on SMBus data line.<br>This register is cleared by HW on reads by FW. |
| 5:3 | 0x0<br>RO | **RSVD1:** Reserved. |
| 2:0 | 0x0<br>RO/V | **Last SMBus State (LSMBS):** This register reflects the last seen state of the SMBus.<br>Since the SMBus state is dynamic, it is advised that FW use its judgment when interpreting this data.<br>• 000: SMBus Idle.<br>• 001: Start Seen.<br>• 010: ACK Seen.<br>• 011: Repeated Start Seen.<br>• 100: Stop Seen.<br>• 101: Reserved.<br>• 110: Reserved.<br>• 111: Reserved |

§ §

# 50 SATA Controllers - B0, D(19, 20), F0

## 50.1 Introduction and Index

The host-accessible registers for the SATA Controllers are described here.

To determine the number of devices/functions available refer to the following:

- Intel Atom® Processor C3000 Product Family
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 50.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 19 (decimal), Function 0. Another set of configuration registers start at Bus 0, Device 20 (decimal), Function 0. The offset addresses are listed.

**Table 50-1. Summary of PCI Configuration Registers—0/19 (or 20)/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 1C028086 | "Identifiers (ID)—Offset 0h" on page 1879 |
| 4 | 2 | 0000 | "Command (CMD)—Offset 4h" on page 1880 |
| 6 | 2 | 02B0 | "Device Status (STS)—Offset 6h" on page 1881 |
| 8 | 1 | 00 | "Revision ID (RID)—Offset 8h" on page 1882 |
| 9 | 1 | 01 | "Programming Interface (PI)—Offset 9h" on page 1882 |
| A | 2 | 0106 | "Class Code (CC)—Offset Ah" on page 1882 |
| 10 | 4 | 00000000 | "MSI-X Table Base Address (MXTBA)—Offset 10h" on page 1883 |
| 18 | 4 | 00000001 | "SCMDBA—Offset 18h" on page 1884 |
| 1C | 4 | 00000001 | "SCTLBA—Offset 1Ch" on page 1884 |
| 20 | 4 | 00000001 | "AHCI Index Data Pair Base Address (AIDPBA)—Offset 20h" on page 1885 |
| 24 | 4 | 00000000 | "AHCI Base Address (ABAR)—Offset 24h" on page 1886 |
| 34 | 1 | 80 | "Capabilities Pointer (CAP)—Offset 34h" on page 1887 |
| 3C | 2 | 0100 | "Interrupt Information (INTR)—Offset 3Ch" on page 1887 |
| 70 | 2 | A801 | "PCI Power Management Capability ID (PID)—Offset 70h" on page 1887 |
| 72 | 2 | 4003 | "PCI Power Management Capabilities (PC)—Offset 72h" on page 1888 |
| 74 | 2 | 0008 | "PCI Power Management Control and Status (PMCS)—Offset 74h" on page 1889 |
| 80 | 2 | 7005 | "Message Signaled Interrupt Identifier (MID)—Offset 80h" on page 1889 |
| 82 | 2 | 0000 | "Message Signaled Interrupt Message Control (MC)—Offset 82h" on page 1890 |
| 84 | 4 | 00000000 | "Message Signaled Interrupt Message Address (MA)—Offset 84h" on page 1890 |
| 88 | 2 | 0000 | "Message Signaled Interrupt Message Data (MD)—Offset 88h" on page 1890 |
| 90 | 4 | 00000000 | "Port Mapping Register (MAP)—Offset 90h" on page 1891 |
| 94 | 4 | 00000000 | "Port Control and Status (PCS)—Offset 94h" on page 1893 |
| 9C | 4 | 00000000 | "SATA General Configuration (SATAGC)—Offset 9Ch" on page 1895 |
| A0 | 1 | 00 | "SATA Initialization Register Index (SIRI)—Offset A0h" on page 1896 |
| A4 | 4 | 00000000 | "SATA Initialization Register Data (SIRD)—Offset A4h" on page 1896 |
| A8 | 4 | 00100012 | "Serial ATA Capability Register 0 (SATACR0)—Offset A8h" on page 1897 |
| AC | 4 | 00000048 | "Serial ATA Capability Register 1 (SATACR1)—Offset ACh" on page 1898 |
| B0 | 2 | 0013 | "FLR Capability ID (FLRCID)—Offset B0h" on page 1898 |
| B4 | 2 | 0000 | "FLR Control (FLRCTL)—Offset B4h" on page 1899 |
| D2 | 2 | 0007 | "MSI-X Message Control (MXC)—Offset D2h" on page 1900 |
| D4 | 4 | 00000000 | "MSI-X Table Offset / Table BIR (MXT)—Offset D4h" on page 1900 |
| D8 | 4 | 00000000 | "MSI-X PBA Offset / PBA BIR (MXP)—Offset D8h" on page 1900 |
| E0 | 4 | 00000000 | "BIST FIS Control/Status (BFCS)—Offset E0h" on page 1901 |

**Table 50-1. Summary of PCI Configuration Registers—0/19 (or 20)/0 (Continued)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| E4 | 4 | 00000000 | "BIST FIS Transmit Data 1 (BFTD1)—Offset E4h" on page 1903 |
| E8 | 4 | 00000000 | "BIST FIS Transmit Data 2 (BFTD2)—Offset E8h" on page 1903 |
| F8 | 4 | 08000FB1 | "Manufacturing ID (MFID)—Offset F8h" on page 1903 |

## 50.1.2 Host Memory Space—MXTBA

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 50-2. Summary of Memory Mapped I/O Registers—MXTBA**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 4 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 8 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 10 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 14 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 18 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 20 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 24 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 28 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 30 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 34 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 38 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 40 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 44 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 48 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 50 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 54 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 58 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 60 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 64 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 68 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |
| 70 | 4 | 00000000 | "MSI-X Table Entries n Message Lower Address (MXTEnMLA)" on page 1904 |
| 74 | 4 | 00000000 | "MSI-X Table Entries n Message Upper Address (MXTEnMUA)" on page 1904 |
| 78 | 4 | 00000000 | "MSI-X Table Entries n Message Data (MXTEnMD)" on page 1905 |

### 50.1.3 Host Memory Space—ABAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 50-3. Summary of Memory Mapped I/O Registers—ABAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | FF36FF07 | "HBA Capabilities (GHC_CAP)—Offset 0h" on page 1906 |
| 4 | 4 | 80000000 | "HBA Capabilities (GHC_CAP)—Offset 0h" on page 1906 |
| 8 | 4 | 00000000 | "Interrupt Status Register (IS)—Offset 8h" on page 1909 |
| C | 4 | 00000000 | "Ports Implemented (GHC_PI)—Offset Ch" on page 1910 |
| 10 | 4 | 00010300 | "AHCI Version (VS)—Offset 10h" on page 1911 |
| 20 | 4 | 07010000 | "Enclosure Management Control (EM_CTL)—Offset 20h" on page 1912 |
| 24 | 4 | 0000003C | "HBA Capabilities Extended (GHC_CAP2)—Offset 24h" on page 1914 |
| A0 | 4 | 00000048 | "Vendor Specific (VSP)—Offset A0h" on page 1915 |
| C0 | 4 | 00311C02 | "RAID Platform ID (RPID)—Offset C0h" on page 1917 |
| C8 | 2 | 003F | "SW Feature Mask (SFM)—Offset C8h" on page 1918 |
| 100 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB0)—Offset 100h" on page 1919 |
| 104 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU0)—Offset 104h" on page 1919 |
| 108 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB0)—Offset 108h" on page 1920 |
| 10C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU0)—Offset 10Ch" on page 1920 |
| 110 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS0)—Offset 110h" on page 1921 |
| 114 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE0)—Offset 114h" on page 1923 |
| 118 | 4 | 00000004 | "Port [0-7] Command (PxCMD0)—Offset 118h" on page 1924 |
| 128 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS0)—Offset 128h" on page 1926 |
| 12C | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL0)—Offset 12Ch" on page 1926 |
| 130 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR0)—Offset 130h" on page 1927 |
| 134 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT0)—Offset 134h" on page 1927 |
| 138 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI0)—Offset 138h" on page 1927 |
| 13C | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF0)—Offset 13Ch" on page 1928 |
| 180 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB1)—Offset 180h" on page 1928 |
| 184 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU1)—Offset 184h" on page 1928 |
| 188 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB1)—Offset 188h" on page 1929 |
| 18C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU1)—Offset 18Ch" on page 1929 |
| 190 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS1)—Offset 190h" on page 1930 |
| 194 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE1)—Offset 194h" on page 1932 |
| 198 | 4 | 00000004 | "Port [0-7] Command (PxCMD1)—Offset 198h" on page 1933 |
| 1A8 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS1)—Offset 1A8h" on page 1935 |
| 1AC | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL1)—Offset 1ACh" on page 1935 |
| 1B0 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR1)—Offset 1B0h" on page 1936 |
| 1B4 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT1)—Offset 1B4h" on page 1936 |
| 1B8 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI1)—Offset 1B8h" on page 1936 |

## Table 50-3. Summary of Memory Mapped I/O Registers—ABAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1BC | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF1)—Offset 1BCh" on page 1937 |
| 200 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB2)—Offset 200h" on page 1937 |
| 204 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU2)—Offset 204h" on page 1937 |
| 208 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB2)—Offset 208h" on page 1938 |
| 20C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU2)—Offset 20Ch" on page 1938 |
| 210 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS2)—Offset 210h" on page 1939 |
| 214 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE2)—Offset 214h" on page 1941 |
| 218 | 4 | 00000004 | "Port [0-7] Command (PxCMD2)—Offset 218h" on page 1942 |
| 228 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS2)—Offset 228h" on page 1944 |
| 22C | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL2)—Offset 22Ch" on page 1944 |
| 230 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR2)—Offset 230h" on page 1945 |
| 234 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT2)—Offset 234h" on page 1945 |
| 238 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI2)—Offset 238h" on page 1945 |
| 23C | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF2)—Offset 23Ch" on page 1946 |
| 280 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB3)—Offset 280h" on page 1946 |
| 284 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU3)—Offset 284h" on page 1946 |
| 288 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB3)—Offset 288h" on page 1947 |

**Table 50-3. Summary of Memory Mapped I/O Registers—ABAR (Continued)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 28C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU3)—Offset 28Ch" on page 1947 |
| 290 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS3)—Offset 290h" on page 1948 |
| 294 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE3)—Offset 294h" on page 1950 |
| 298 | 4 | 00000004 | "Port [0-7] Command (PxCMD3)—Offset 298h" on page 1951 |
| 2A8 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS3)—Offset 2A8h" on page 1953 |
| 2AC | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL3)—Offset 2ACh" on page 1953 |

## Table 50-3. Summary of Memory Mapped I/O Registers—ABAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2B0 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR3)—Offset 2B0h" on page 1954 |
| 2B4 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT3)—Offset 2B4h" on page 1954 |
| 2B8 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI3)—Offset 2B8h" on page 1954 |
| 2BC | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF3)—Offset 2BCh" on page 1955 |
| 300 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB4)—Offset 300h" on page 1955 |
| 304 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU4)—Offset 304h" on page 1955 |
| 308 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB4)—Offset 308h" on page 1956 |
| 30C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU4)—Offset 30Ch" on page 1956 |
| 310 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS4)—Offset 310h" on page 1957 |
| 314 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE4)—Offset 314h" on page 1959 |
| 318 | 4 | 00000004 | "Port [0-7] Command (PxCMD4)—Offset 318h" on page 1960 |
| 328 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS4)—Offset 328h" on page 1962 |
| 32C | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL4)—Offset 32Ch" on page 1962 |
| 330 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR4)—Offset 330h" on page 1963 |
| 334 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT4)—Offset 334h" on page 1963 |
| 338 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI4)—Offset 338h" on page 1963 |
| 33C | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF4)—Offset 33Ch" on page 1964 |
| 380 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB5)—Offset 380h" on page 1964 |
| 384 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU5)—Offset 384h" on page 1964 |
| 388 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB5)—Offset 388h" on page 1965 |
| 38C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU5)—Offset 38Ch" on page 1965 |
| 390 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS5)—Offset 390h" on page 1966 |
| 394 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE5)—Offset 394h" on page 1968 |
| 3A8 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS5)—Offset 3A8h" on page 1969 |
| 3AC | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL5)—Offset 3ACh" on page 1969 |
| 3B0 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR5)—Offset 3B0h" on page 1970 |
| 3B4 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT5)—Offset 3B4h" on page 1970 |
| 3B8 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI5)—Offset 3B8h" on page 1970 |
| 3BC | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF5)—Offset 3BCh" on page 1971 |
| 3C4 | 4 | 1E022852 | "Port [0-7] Device Sleep (PxDEVSLP5)—Offset 3C4h" on page 1972 |
| 400 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB6)—Offset 400h" on page 1972 |
| 404 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU6)—Offset 404h" on page 1973 |
| 408 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB6)—Offset 408h" on page 1973 |
| 40C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU6)—Offset 40Ch" on page 1973 |
| 410 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS6)—Offset 410h" on page 1974 |
| 414 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE6)—Offset 414h" on page 1976 |
| 418 | 4 | 00000004 | "Port [0-7] Command (PxCMD6)—Offset 418h" on page 1977 |

## Table 50-3. Summary of Memory Mapped I/O Registers—ABAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 420 | 4 | 0000007F | "Port [0-7] Task File Data (PxTFD6)—Offset 420h" on page 1979 |
| 424 | 4 | FFFFFFFF | "Port [0-7] Signature (PxSIG6)—Offset 424h" on page 1979 |
| 428 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS6)—Offset 428h" on page 1980 |
| 42C | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL6)—Offset 42Ch" on page 1980 |
| 430 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR6)—Offset 430h" on page 1981 |
| 434 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT6)—Offset 434h" on page 1981 |
| 438 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI6)—Offset 438h" on page 1981 |
| 43C | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF6)—Offset 43Ch" on page 1982 |
| 444 | 4 | 1E022852 | "Port [0-7] Device Sleep (PxDEVSLP6)—Offset 444h" on page 1982 |
| 480 | 4 | 00000000 | "Port [0-7] Command List Base Address (PxCLB7)—Offset 480h" on page 1983 |
| 484 | 4 | 00000000 | "Port [0-7] Command List Base Address Upper 32-bits (PxCLBU7)—Offset 484h" on page 1983 |
| 488 | 4 | 00000000 | "Port [0-7] FIS Base Address (PxFB7)—Offset 488h" on page 1984 |
| 48C | 4 | 00000000 | "Port [0-7] FIS Base Address Upper 32-bits (PxFBU7)—Offset 48Ch" on page 1984 |
| 490 | 4 | 00000000 | "Port [0-7] Interrupt Status (PxIS7)—Offset 490h" on page 1985 |
| 494 | 4 | 00000000 | "Port [0-7] Interrupt Enable (PxIE7)—Offset 494h" on page 1987 |
| 498 | 4 | 00000004 | "Port [0-7] Command (PxCMD7)—Offset 498h" on page 1988 |
| 4A0 | 4 | 0000007F | "Port [0-7] Task File Data (PxTFD7)—Offset 4A0h" on page 1990 |
| 4A4 | 4 | FFFFFFFF | "Port [0-7] Signature (PxSIG7)—Offset 4A4h" on page 1990 |
| 4A8 | 4 | 00000000 | "Port [0-7] Serial ATA Status (PxSSTS7)—Offset 4A8h" on page 1991 |
| 4AC | 4 | 00000000 | "Port [0-7] Serial ATA Control (PxSCTL7)—Offset 4ACh" on page 1991 |
| 4B0 | 4 | 00000000 | "Port [0-7] Serial ATA Error (PxSERR7)—Offset 4B0h" on page 1992 |
| 4B4 | 4 | 00000000 | "Port [0-7] Serial ATA Active (PxSACT7)—Offset 4B4h" on page 1992 |
| 4B8 | 4 | 00000000 | "Port [0-7] Commands Issued (PxCI7)—Offset 4B8h" on page 1992 |
| 4BC | 4 | 00000000 | "Port [0-7] SNotification (PxSNTF7)—Offset 4BCh" on page 1993 |
| 580 | 4 | 00000000 | "Enclosure Management Message Format (EM_MF)—Offset 580h" on page 1993 |
| 584 | 4 | 00000000 | "Enclosure Management LED (EM_LED)—Offset 584h" on page 1994 |

## 50.2 Registers in Configuration Space

### 50.2.1 Identifiers (ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 0h

**Default:** 1C028086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x1c02<br>RO | **Device ID (DID):** The specific value is dependent on config bits and fuses. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates the company vendor as Intel. |

## 50.2.2 Command (CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved. |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** This disables pin-based INTx# interrupts.<br>This bit has no effect on MSI operation. When set, internal INTx# will not be generated.<br>When cleared, internal INTx# are generated if there is an interrupt and MSI is not enabled. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Reserved. |
| 8 | 0x0<br>RW | **SERR# Enable (SEE):** When set to 1, the HBA is allowed to generate SERR# on DPD or SATAGC.URD event that is enabled for SERR# generation.<br>When cleared to 0, it is not. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Reserved. |
| 6 | 0x0<br>RW | **Parity Error Response Enable (PEE):** When set, the SATA Controller will corrupt the outbound DATA FIS CRC if a forwarded data parity error is indicated. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VGA):** Reserved. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Reserved. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Reserved. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the SATA Controller's ability to act as a master for data transfers.<br>This bit does not impact the generation of completions for split transaction commands. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the SATA Controller's target memory space (for AHCI). |
| 0 | 0x0<br>RW | **I/O Space Enable (IOSE):** Controls access to the SATA Controller's target I/O space. |

## 50.2.3 Device Status (STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 6h

**Default:** 02B0h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **Detected Parity Error (DPE):** Set when the SATA Controller detects a parity error on its interface. |
| 14 | 0x0<br>RW/1C | **Signaled System Error (SSE):** Set when SATA Controller generates an SERR#. |
| 13 | 0x0<br>RW/1C | **Received Master-Abort Status (RMA):** Set when the SATA Controller receives a master abort to a cycle it generated. |
| 12 | 0x0<br>RW/1C | **Received Target-Abort Status (RTA):** Set when the SATA Controller receives a target abort to a cycle it generated. |
| 11 | 0x0<br>RW/1C | **Signaled Target-Abort Status (STA):** This bit must be set by a target device whenever it terminates a transaction with Target-Abort.<br>Devices that will never signal Target-Abort do not need to implement this bit. |
| 10:9 | 0x1<br>RO | **DEVSEL# Timing Status (DEVT):** Controls the device select time for the SATA Controller's PCI interface. |
| 8 | 0x0<br>RW/1C | **Master Data Parity Error Detected (DPD):** Set when the SATA Controller, as a master, either detects a parity error or sees the parity error line asserted, and the parity error response bit (bit 6 of the command register) is set.<br>This bit can only be set on read completions received from the backbone where there is a parity error. |
| 7 | 0x1<br>RO | **Fast Back-to-Back Capable (FBC):** Reserved. |
| 6 | 0x0<br>RO | **RSVD0:** Reserved. |
| 5 | 0x1<br>RO | **66 MHz Capable (RSV):** Reserved. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list. |
| 3 | 0x0<br>RO | **Interrupt Status (IS):** Reflects the state of INTx# messages, IRQ14 or IRQ15.<br>This bit is set when the interrupt is to be asserted.<br>This bit is a 0 after the interrupt is cleared. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved. |

## 50.2.4 Revision ID (RID)—Offset 8h

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Revision ID (RID):** Indicates stepping of the host controller hardware. |

## 50.2.5 Programming Interface (PI)—Offset 9h

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 9h

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RO | **Interface (IF):** If CC.SCC=06h(AHCI mode), it indicates that this is an AHCI HBA that has a major revision of 1 (as specified in the AHCI Version register).<br>If CC.SCC=04h(RAID mode), it indicates that there is no programming interface(IF=00h).<br>Internally, under this condition, the SATA controller is in native mode and its I/O spaces are only accessible through the I/O BARs. |

## 50.2.6 Class Code (CC)—Offset Ah

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + Ah

**Default:** 0106h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO | **Base Class Code (BCC):** Indicates that this is a mass storage device. |
| 7:0 | 0x6<br>RO | **Sub Class Code (SCC):** The value reported in this field is dependent on SATAGC.SMS and various fuses and configuration bits. |

## 50.2.7 MSI-X Table Base Address (MXTBA)—Offset 10h

This BAR is used to allocate 32K, 16K or 8K Memory space for the MSI-X Table. The Memory space size is determined by BIOS by making bit-14 and bit-13 Read-Only '1' or Read-Write '0' based on SATAGC.MSS[1:0].

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 10h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:15 | 0x0<br>RW | **Base Address (BA):** Base address of memory space. |
| 14 | 0x0<br>RW | **Base Address Bit 14 (BAB14):** When SATAGC.MSS [1:0]=00, this bit is Read Only '0' else it's Read Write '0'. |
| 13 | 0x0<br>RW | **Base Address Bit 13 (BAB13):** When SATAGC.MSS [1:0]=00 or 01, this bit is Read Only '0' else it's Read Write '0'. |
| 12:4 | 0x0<br>RO | **RSVD0:** Reserved. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not prefetchable. |
| 2:1 | 0x0<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 32-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for Memory space. |

## 50.2.8 SCMDBA—Offset 18h

BAR2 register define a 8-byte I/O space but this space is not implemented. Accesses to this space will get master abort.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 18h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:3 | 0x0<br>RW | **Base Address (BAR):** Base address of the I/O space. |
| 2:1 | 0x0<br>RO | **RSVD1:** Reserved. |
| 0 | 0x1<br>RO | **Resource Type Indicator (RTE):** Indicates a request for IO space. |

## 50.2.9 SCTLBA—Offset 1Ch

BAR3 register define a 4-byte I/O space but this space is not implemented. Accesses to this space will get master abort.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 1Ch

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:2 | 0x0<br>RW | **Base Address (BAR):** Base address of the I/O space. |
| 1 | 0x0<br>RO | **RSVD1:** Reserved. |
| 0 | 0x1<br>RO | **Resource Type Indicator (RTE):** Indicates a request for IO space. |

## 50.2.10 AHCI Index Data Pair Base Address (AIDPBA)—Offset 20h

This BAR is used to allocate I/O space for the AHCI index/data pair mechanism.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 20h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:5 | 0x0<br>RW | **Base Address (BA):** Base address of the I/O space. |
| 4:1 | 0x0<br>RO | **RSVD1:** Reserved. |
| 0 | 0x1<br>RO | **Resource Type Indicator (RTE):** Indicates a request for IO space. |

## 50.2.11 AHCI Base Address (ABAR)—Offset 24h

This register represents a memory BAR allocating space for the AHCI memory registers.

*Note:* Bit [31:16] of this register must be programmed to a value greater than 0 to ensure the memory is mapped to an address of 1 Mbyte and greater (i.e. ABAR must be 00010000h or greater). Otherwise, memory cycle targeting the ABAR range may not be accepted.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 24h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RW | **Base Address (BA):** Base address of register memory space. |
| 18 | 0x0<br>RW | **Base Address Bit 18 (BAB18):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 256K, this bit is Read Only '0' else it's Read Write '0'. |
| 17 | 0x0<br>RW | **Base Address Bit 17 (BAB17):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 128K, this bit is Read Only '0' else it's Read Write '0'. |
| 16 | 0x0<br>RW | **Base Address Bit 16 (BAB16):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 64K, this bit is Read Only '0' else it's Read Write '0'. |
| 15 | 0x0<br>RW | **Base Address Bit 15 (BAB15):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 32K, this bit is Read Only '0' else it's Read Write '0'. |
| 14 | 0x0<br>RW | **Base Address Bit 14 (BAB14):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 16K, this bit is Read Only '0' else it's Read Write '0'. |
| 13:11 | 0x0<br>RW | **Base Address Bit 13-11 (BAB1311):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 2K, this bit is Read Only '0' else it's Read Write '0'. |
| 10:4 | 0x0<br>RO | **RSVD0:** Reserved. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not prefetchable |
| 2:1 | 0x0<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 32-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

## 50.2.12 Capabilities Pointer (CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 34h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x80<br>RW/L | **Capability Pointer (CP):** Indicates that the first capability pointer offset is offset 80h (the Message Signaled Interrupt capability).<br>The following capability structures are linked by default:<br>• (CAP.CP -) 80h<br>• (MSI) -) D0h<br>• (MSI-X)-) 70h<br>• (PCI Power) -) A8h<br>• (SATA) -) 00h end. |

## 50.2.13 Interrupt Information (INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RW/O | **Interrupt Pin (IPIN):** This register tells which interrupt pin the device function uses.<br>• A value of 1 corresponds to INTA#.<br>• A value of 2 corresponds to INTB#.<br>• A value of 3 corresponds to INTC#.<br>• A value of 4 corresponds to INTD#.<br>This register is not reset by FLR. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this register.<br>Interrupt Line register is not reset by FLR. |

## 50.2.14 PCI Power Management Capability ID (PID)—Offset 70h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 70h

**Default:** A801h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0xa8<br>RW/L | **Next Capability (NEXT):** A8h is location of the Serial ATA Capability structure.<br>The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function.<br>Refer to SATAGC.REGLOCK description in order to lock the register to become RO.<br>This register is not reset by FLR. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management capability. |

## 50.2.15 PCI Power Management Capabilities (PC)—Offset 72h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 72h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME_Support:** The default value is 01000 which indicates PME# can be generated from the D3HOT state in the SATA controller. |
| 10 | 0x0<br>RO | **D2_Support:** The D2 state is not supported. |
| 9 | 0x0<br>RO | **D1_Support:** The D1 state is not supported. |
| 8:6 | 0x0<br>RO | **Aux_Current:** PME# from D3COLD state is not supported, therefore this field is 000b. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates that no device-specific initialization is required. |
| 4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 50.2.16 PCI Power Management Control and Status (PMCS)—Offset 74h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 74h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **PME Status (PMES):** This bit is set when a PME event is to be requested, and if this bit is set and PMEE is set, a PME# will be generated.<br>This register field is not reset by FLR. |
| 14:9 | 0x0<br>RO | **RSVD0:** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, the SATA controller generates PME# from D3HOT on a wake event.<br>Note: Software is advised to clear PMEE together with PMES prior to changing CC.SCC thru SATAGC.SMS.<br>This register field is not reset by FLR. |
| 7:4 | 0x0<br>RO | **RSVD1:** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSFRST):** A 1 indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br>Configuration context is preserved. |
| 2 | 0x0<br>RO | **RSVD2:** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the SATA Controller and to set a new power state.<br>The values are:<br>• 00 = D0 state.<br>• 11 = D3HOT state. |

## 50.2.17 Message Signaled Interrupt Identifier (MID)—Offset 80h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 80h

**Default:** 7005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x70<br>RW/L | **Next Pointer (NEXT):** Indicates the next item in the list is the PCI power management pointer. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Capabilities ID indicates MSI. |

## 50.2.18 Message Signaled Interrupt Message Control (MC)—Offset 82h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 82h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved. |
| 7 | 0x0<br>RO | **64 Bit Address Capable (C64):** Capable of generating a 32-bit message only. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** When this field is cleared to 000 (and MSIE is set), only a single MSI message will be generated for all SATA ports, and bits [15:0] of the message vector will be driven from MD[15:0]. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not supported. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. |

## 50.2.19 Message Signaled Interrupt Message Address (MA)—Offset 84h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DWORD aligned. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 50.2.20 Message Signaled Interrupt Message Data (MD)—Offset 88h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 88h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the lower word of the data bus of the MSI memory write transaction. |

## 50.2.21 Port Mapping Register (MAP)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23 | 0x0<br>RW/O | **SATA Port 7 Disable (SPD7):** Similar to SPD0 but for port 7.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>PCIe/SATA muxing for port 7 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 7 physically. |
| 22 | 0x0<br>RW/O | **SATA Port 6 Disable (SPD6):** Similar to SPD0 but for port 6.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>PCIe/SATA muxing for port 6 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 6 physically. |
| 21 | 0x0<br>RW/O | **SATA Port 5 Disable (SPD5):** Similar to SPD0 but for port 5.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>PCIe/SATA muxing for port 5 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 5 physically. |
| 20 | 0x0<br>RW/O | **SATA Port 4 Disable (SPD4):** Similar to SPD0 but for port 4.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>PCIe/SATA muxing for port 4 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 4 physically. |
| 19 | 0x0<br>RW/O | **SATA Port 3 Disable (SPD3):** Similar to SPD0 but for port 3.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>• 1. Fuse FFSATA7 (disable port 2 and 3).<br>• 2. Fuse FFSATA8 (disable port 1 and 3).<br>• 3. PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 3 physically. |
| 18 | 0x0<br>RW/O | **SATA Port 2 Disable (SPD2):** Similar to SPD0 but for port 2.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>• 1. Fuse FFSATA7 (disable port 2 and 3).<br>• 2. PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 3 physically. |
| 17 | 0x0<br>RW/O | **SATA Port 1 Disable (SPD1):** Similar to SPD0 but for port 1.<br>This bit is RO-one if ANY of the fuse/strap/GPIO condition below is true else this bit is RW-zero.<br>• 1. Fuse FFSATA8 (disable port 1 and 3).<br>• 2. PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA.<br>This bit is only applicable to project(s): that has port 1 physically. |
| 16 | 0x0<br>RW/O | **SATA Port 0 Disable (SPD0):** A 1 prevents the SATA port from being enabled via config PCS.PxE. Write of 1 to PCS.PxE has no effect when the corresponding SPD[x] bit is 1.<br>In preventing a port(s) from being enabled, BIOS shall first configure MAP.SPDx. And only then BIOS configures the PCS.PxE.<br>This field is not reset by FLR.<br>This bit is only applicable to project(s): that has port 0 physically. |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Port Clock Disable (PCD):** When any of these bits is set to 1, the backbone clock driven to the associated port logic is gated and will not toggle.<br>When this bit is cleared to 0, all clocks to the associated port logic will operate normally.<br>Assignment of the bits is:<br>• Bit 7: Port 7, this bit is applicable to projects: that has port 7 physically<br>• Bit 6: Port 6, this bit is only applicable to project(s): that has port 6 physically<br>• Bit 5: Port 5, this bit is only applicable to project(s): that has port 5 physically<br>• Bit 4: Port 4, this bit is only applicable to project(s): that has port 4 physically<br>• Bit 3: Port 3, this bit is only applicable to project(s): that has port 3 physically<br>• Bit 2: Port 2, this bit is only applicable to project(s): that has port 2 physically<br>• Bit 1: Port 1, this bit is only applicable to project(s): that has port 1 physically<br>• Bit 0: Port 0, this bit is only applicable to project(s): That has port 0 physically.<br>If a particular port is not available, software shall set the corresponding bit to 1.<br>Software can also set the corresponding bit(s) to 1 after disabling particular port(s).<br>Software cannot set the PCD[port x]=1 if the corresponding config PCS.PxE=1 or AHCI MMIO GHC.PI[x]=1. |

## 50.2.22 Port Control and Status (PCS)—Offset 94h

By default, the SATA ports are set (by hardware) to the disabled state (e.g. bits[5:0] == '0') as a result of an initial power on reset. When enabled by software, the ports can transition between the on, partial, and slumber states and can detect devices. When disabled, the port is in the off state and cannot detect any devices. This register is not reset by FLR.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23 | 0x0<br>RO | **Port 7 Present (P7P):** Same as P0P, except for port 7.<br>This bit is only applicable to project(s): That has port 7 physically. |
| 22 | 0x0<br>RO | **Port 6 Present (P6P):** Same as P0P, except for port 6.<br>This bit is only applicable to project(s): That has port 6 physically. |
| 21 | 0x0<br>RO | **Port 5 Present (P5P):** Same as P0P, except for port 5.<br>This bit is only applicable to project(s): That has port 5 physically. |
| 20 | 0x0<br>RO | **Port 4 Present (P4P):** Same as P0P, except for port 4.<br>This bit is only applicable to project(s): That has port 4 physically. |
| 19 | 0x0<br>RO | **Port 3 Present (P3P):** Same as P0P, except for port 3.<br>This bit is only applicable to project(s): That has port 3 physically. |
| 18 | 0x0<br>RO | **Port 2 Present (P2P):** Same as P0P, except for port 2.<br>This bit is only applicable to project(s): That has port 2 physically. |
| 17 | 0x0<br>RO | **Port 1 Present (P1P):** Same as P0P, except for port 1.<br>This bit is only applicable to project(s): That has port 1 physically. |
| 16 | 0x0<br>RO | **Port 0 Present (P0P):** When set, the SATA controller has detected the presence of a device on port 0. It may change at any time. Clearing P0E bit leads to clearing of this bit after implementation delay.<br>***Note:*** For system software that intends to clear all PCS.PxE bits that are previously 1 to 0 and then to 1 again in two consecutive write cycles, software shall poll on this bit being 0 before setting P0E bit to 1. This bit is only applicable to project(s): That has port 0 physically. |
| 15:8 | 0x0<br>RO | **RSVD1:** Reserved. |
| 7 | 0x0<br>RW | **Port 7 Enabled (P7E):** When MAP.SPD[7] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 7.<br>This bit takes precedence over P7CMD.SUD.<br>This bit is only applicable to project(s): That has port 7 physically. |
| 6 | 0x0<br>RW | **Port 6 Enabled (P6E):** When MAP.SPD[6] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 6.<br>This bit takes precedence over P6CMD.SUD.<br>This bit is only applicable to project(s): That has port 6 physically. |
| 5 | 0x0<br>RW | **Port 5 Enabled (P5E):** When MAP.SPD[5] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 5.<br>This bit takes precedence over P5CMD.SUD.<br>This bit is only applicable to project(s): That has port 5 physically. |
| 4 | 0x0<br>RW | **Port 4 Enabled (P4E):** When MAP.SPD[4] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this bit is RW and same as P0E but for port 4 and takes precedence over P4CMD.SUD.<br>This bit is only applicable to project(s): That has port 4 physically. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW | **Port 3 Enabled (P3E):** When MAP.SPD[3] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this bit is RW and same as P0E but for port 3 and takes precedence over P3CMD.SUD.<br>This bit is only applicable to project(s): That has port 3 physically. |
| 2 | 0x0<br>RW | **Port 2 Enabled (P2E):** When MAP.SPD[2] is 1, this bit is reserved and is read-only 0.<br>Otherwise if none of the above is true, this bit is RW and same as P0E but for port 2 and takes precedence over P2CMD.SUD.<br>This bit is only applicable to project(s): That has port 2 physically. |
| 1 | 0x0<br>RW | **Port 1 Enabled (P1E):** When MAP.SPD[1] is 1, this bit is reserved and read-only 0.<br>Otherwise if none of the above is true, this bit is RW and same as P0E but for port 1 and takes precedence over P1CMD.SUD.<br>This bit is only applicable to project(s): That has port 1 physically. |
| 0 | 0x0<br>RW | **Port 0 Enabled (P0E):** When MAP.SPD[0] is 1, this bit is reserved and read-only 0.<br>When set, the port is enabled.<br>When cleared, the port is disabled.<br>When enabled, the port can transition between the on, partial, and slumber states and can detect devices.<br>When disabled, the port is in the off state and cannot detect any devices.<br>This bit takes precedence over P0CMD.SUD. |

## 50.2.23 SATA General Configuration (SATAGC)—Offset 9Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/O | **Register Lock (REGLOCK):** BIOS can set this bit to 1 to lock the following registers with RW/L attribute:<br>• CAP<br>• CP<br>• MID.NEXT<br>• PIE.NEXT<br>• SATACR0.NEXT.<br>Once locked the register attribute of above list changes from RW/L to RO holding the existing value.<br>This field is not reset by FLR. |
| 30:17 | 0x0<br>RO | **RSVD0:** Reserved. |
| 16 | 0x0<br>RW | **SATA Mode Select (SMS):** Software (SW) programs these bits to control the mode in which the SATA HBA should operate:<br>• 0b = AHCI mode.<br>• 1b = RAID mode.<br>This register field is not reset by FLR. |
| 15 | 0x0<br>RW | **Data Phase Parity Error Enable (DPPEE):**<br>When 1, IOSF data phase parity error handling is enabled.<br>When 0, the data phase parity error handling is disabled. |
| 14:12 | 0x0<br>RW | **Write Request_Size Select/Max_Payload_Size (WRRSELMPS):** These two bits select the max write request size that SATA host controller will initiate for DMA write to memory.<br>SATA host controller will internally break up larger write request based on these bits.<br>The request is address-aligned to the selected size.<br>Defined encodings for this field are:<br>• 000b = 128 address aligned bytes max payload size.<br>• 111b = 64 address aligned bytes max payload size.<br>• All other values are reserved for SATA host controller.<br>This field is not reset by FLR. |
| 11 | 0x0<br>RW | **Command Parity Error Enable (CPEE):**<br>When 1, command parity error handing is enable.<br>When 0 the command parity error handling is disable |
| 10 | 0x0<br>RW | **SATA Controller Function Disable (SCFD):** BIOS program this bit to 1 to disable the SATA Controller function.<br>When 0, SATA Controller function is enabled. |
| 9 | 0x0<br>RW | **Unsupported Request Reporting Enable (URRE):** If set to 1 by software, it allows reporting of an Unsupported Request as a system error.<br>If both URRE and PCI configuration SERR# Enable registers are set to a 1, then the agent must set the Signaled System Error bit in the PCI Status register and send a DO_SERR message in IOSF-SB interface. |
| 8 | 0x0<br>RW/1C | **Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unsupported Request on IOSF Primary interface that is not considered Advisory Non-Fatal.<br>Cleared to 0 by SW. |
| 7 | 0x0<br>RW/O | **Alternate ID Enable (AIE):** When programmed to 0, HW will report the following device id's: 2822h for desktop or 282Ah for mobile.<br>When programmed to a 1, HW will not report these device id's.<br>This field is not reset by FLR. |
| 6 | 0x0<br>RW/O | **AIE0 DevID Selection (DEVIDSEL):** This register allows BIOS to select Device ID when AIE=0 and Server Feature (SATA AIE DEVIDSEL) Disable Fuse =0.<br>This field is not reset by FLR. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW/O | **FLR Capability Selection (FLRCSSEL):** This allows the FLR Capability to be bypassed. |
| 4:3 | 0x0<br>RW/O | **MXTBA Size Select (MSS):** These 2 bits select the size of the Memory space for the MSI-X Table defined in BAR 0 (Configuration space offset 10h).<br>MSI-X Table Memory space size is:<br>• 32k when MSS[1:0]=00.<br>• 16k when MSS[1:0]=01.<br>• 8k when MSS[1:0]=10.<br>This field is not reset by FLR. |
| 2:0 | 0x0<br>RW/O | **ABAR Size Select (ASSEL):** These 3 bits select the size of the Memory space for the ABAR in BAR 5 (Configuration space offset 24h).<br>ABAR Memory space size is:<br>• 2k when ASSEL[2:0]=000.<br>• 16k when ASSEL[2:0]=001.<br>• 32k when ASSEL[2:0]=010.<br>• 64k when ASSEL[2:0]=011.<br>• 128k when ASSEL[2:0]=100.<br>• 256k when ASSEL[2:0]=101.<br>• 512k when ASSEL[2:0]=110.<br>This field is not reset by FLR. |

## 50.2.24 SATA Initialization Register Index (SIRI)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:19 or 20, F:0] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:2 | 0x0<br>RW | **Index (IDX):** 6-bit index pointer into the 256-byte space.<br>Data is written into the SIRD register and read from the SIRD register.<br>This points to a DWord register.<br>The byte enables on the SIRD register affect what will be written. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved. |

## 50.2.25 SATA Initialization Register Data (SIRD)—Offset A4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + A4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Data (DTA):** 32-bit data value that is written to the register pointed to by SIRI, or read from the register pointed to by SIRI. |

## 50.2.26 Serial ATA Capability Register 0 (SATACR0)—Offset A8h

Note that the SATACR0.NEXT is not changed from RO to become RWO because there is an existing method (SATAGC.FLRCSSEL bit) to bypass the FLR Capability structure.

FLR Capability ID.NEXT is already indicating end of capability structure, it does not need to change to be RWO.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + A8h

**Default:** 00100012h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved. |
| 23:20 | 0x1<br>RO | **Major Revision (MAJREV):** Major revision number of the SATA Capability Pointer implemented. |
| 19:16 | 0x0<br>RO | **Minor Revision (MINREV):** Minor revision number of the SATA Capability Pointer implemented. |
| 15:8 | 0x0<br>RW/L | **Next Capability Pointer (NEXT):** 00h indicating the final item in the Capability List.<br>The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. |
| 7:0 | 0x12<br>RO | **Capability ID (CAP):** The value of 12h has been assigned by the PCI SIG to designate the SATA Capability pointer. |

## 50.2.27 Serial ATA Capability Register 1 (SATACR1)—Offset ACh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + ACh

**Default:** 00000048h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved. |
| 15:4 | 0x4<br>RO | **BAR Offset (BAROFST):** Indicates the offset into the BAR where the AHCI Index/Data pair are located (in Dword granularity).<br>The Index and Data I/O registers are located at offset 10h within the I/O space defined by LBAR (BAR4).<br>A value of 004h indicates offset 10h.<br>• 000h = 0h offset.<br>• 001h = 4h offset.<br>• 002h = 8h offset.<br>• 003h = Ch offset.<br>• 004h = 10h offset.<br>• ...<br>• FFFh = 3FFFh offset (max 16 KB). |
| 3:0 | 0x8<br>RO | **BAR Location (BARLOC):** Indicates the absolute PCI Configuration Register address of the BAR containing the Index/Data pair (in Dword granularity).<br>The Index and Data I/O registers reside within the space defined by LBAR (BAR4) in the SATA controller.<br>A value of 8h indicates offset 20h, which is LBAR (BAR4).<br>• 0000 - 0011b = Reserved<br>• 0100b = 10h =) BAR0<br>• 0101b = 14h =) BAR1<br>• 0110b = 18h =) BAR2<br>• 0111b = 1Ch =) BAR3<br>• 1000b = 20h =) AIDPBA<br>• 1001b = 24h =) BAR5<br>• 1010 - 1110b = Reserved<br>• 1111b = Index/Data pair in PCI Configuration space which is not supported. |

## 50.2.28 FLR Capability ID (FLRCID)—Offset B0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + B0h

**Default:** 0013h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Capability (NEXT):** 00h indicating the final item in the Capability List. |
| 7:0 | 0x13<br>RO | **Capability ID (CID):** The value of this field depends on the FLRCSSEL bit:<br>• SATAGC.FLRCSSEL = 0, Capability ID = 13h.<br>• SATAGC.FLRCSSEL = 1, Capability ID = 00h (capability is bypassed). |

## 50.2.29 FLR Control (FLRCTL)—Offset B4h

This register shall be read-only 0 when SATAGC.FLRCSSEL=1.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + B4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **RSVD0:** Reserved. |
| 8 | 0x0<br>RO | **Transactions Pending (TXP):**<br>A 1 indicates that the controller has issued Non-Posted request which has not been completed.<br>A 0 indicates that Completions for all non-posted requests have been received by the controller. |
| 7:1 | 0x0<br>RO | **RSVD1:** Reserved. |
| 0 | 0x0<br>RW | **Initiate FLR (Initiate_FLR):** Used to initiated FLR transition.<br>A write of 1 indicates FLR transition.<br>Since hardware must not respond to any cycles till FLR completion, the value read by software from this bit is 0.<br>Refer to Function Level Reset (FLR) on specifics to SATA. |

## 50.2.30 MSI-X Message Control (MXC)—Offset D2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:19 or 20, F:0] + D2h

**Default:** 0007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **MSI-X Enable (MXE):** If set to '1' and the MSI Enable bit in the MSI Message Control register is cleared to '0', the function is permitted to use MSI-X to request service and is prohibited from using its INTx# pin (if implemented).<br>If cleared to '0', the function is prohibited from using MSI-X to request service. |
| 14 | 0x0<br>RW | **Function Mask (FM):**<br>If set to '1', all of the vectors associated with the function are masked, regardless of their per vector Mask bit states.<br>If cleared to '0', each vector's Mask bit determines whether the vector is masked or not.<br>Setting or clearing the MSI-X Function Mask bit has no effect on the state of the per vector Mask bits. |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved. |
| 10:0 | 0x7<br>RO | **Table Size (TS):** This value indicates the size of the MSI-X Table as the value n, which is encoded as n - 1. |

## 50.2.31 MSI-X Table Offset / Table BIR (MXT)—Offset D4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Table Offset (TO):** Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X Table.<br>The lower three Table BIR bits are masked off (cleared to 000b) by system software to form a 32-bit Qword-aligned offset. |
| 2:0 | 0x0<br>RO | **Table BIR (TBIR):** This field indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into system memory.<br>A read-only value of '0' means 10h. |

## 50.2.32 MSI-X PBA Offset / PBA BIR (MXP)—Offset D8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **PBA Offset (PBAO):** Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X PBA.<br>The lower three PBA BIR bits are masked off (cleared to 000b) by software to form a 32-bit Qword-aligned offset. |
| 2:0 | 0x0<br>RO | **PBA BIR (PBIR):** This field indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X PBA into system memory.<br>A read-only value of '1' means 14h. |

## 50.2.33 BIST FIS Control/Status (BFCS)—Offset E0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + E0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **RSVD0:** Reserved |
| 17 | 0x0<br>RW | **Port 7 BIST FIS Initiate (P7BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 7, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 7 physically. |
| 16 | 0x0<br>RW | **Port 6 BIST FIS Initiate (P6BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 6, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 6 physically. |
| 15 | 0x0<br>RW | **Port 5 BIST FIS Initiate (P5BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 5, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 5 physically. |
| 14 | 0x0<br>RW | **Port 4 BIST FIS Initiate (P4BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 4, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 4 physically. |
| 13 | 0x0<br>RW | **Port 3 BIST FIS Initiate (P3BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 3, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 3 physically. |
| 12 | 0x0<br>RW | **Port 2 BIST FIS Initiate (P2BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 2, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 2 physically. |
| 11 | 0x0<br>RW/1C/V | **BIST FIS Successful (BFS):** This bit is set any time a BIST FIS transmitted by the SATA controller receives an R_OK completion status from the device. |
| 10 | 0x0<br>RW/1C/V | **BIST FIS Failed (BFF):** This bit is set any time that a BIST FIS transmitted by the SATA controller receives an R_ERR completion status from the device. |
| 9 | 0x0<br>RW | **Port 1 BIST FIS Initiate (P1BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 1, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 1 physically. |
| 8 | 0x0<br>RW | **Port 0 BIST FIS Initiate (P0BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 0, using the parameters specified in this register and BFTD1 and BFTD2.<br>This bit is only applicable to project(s): That has port 0 physically. |
| 7:2 | 0x0<br>RW | **BIST FIS Parameters (BFP):** These bits form the contents of the upper 6 bits of the BIST FIS Pattern Definition in the BIST FIS transmitted by the SATA controller.<br>This field is not port specific - its contents will be used for any BIST FIS initiated on the SATA controller.<br>The specific bit definitions are:<br>• Bit 7 (T) Far End Transmit mode.<br>• Bit 6 (A) Align Bypass mode.<br>• Bit 5 (S) Bypass Scrambling.<br>• Bit 4 (L) Far End Retimed Loopback.<br>• Bit 3 (F) Far End Analog Loopback.<br>• Bit 2 (P) Primitive bit for use with Transmit mode. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1:0 | 0x0<br>RO | **RSVD1:** Reserved. |

## 50.2.34 BIST FIS Transmit Data 1 (BFTD1)—Offset E4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + E4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Data (DATA):** The data programmed into this register will form the contents of the second DW of any BIST FIS initiated by the SATA controller.<br>This register is not port specific - its contents will be used for BIST FIS initiated on any port.<br>Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the T bit of the BIST FIS is set to indicate Far-End Transmit mode, this register's contents will be transmitted as the BIST FIS 2nd DW regardless of whether or not the T bit is indicated in the BFCS register. |

## 50.2.35 BIST FIS Transmit Data 2 (BFTD2)—Offset E8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + E8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Data (DATA):** The data programmed into this register will form the contents of the third DW of any BIST FIS initiated by the SATA controller.<br>This register is not port specific - its contents will be used for BIST FIS initiated on any port.<br>Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the T bit of the BIST FIS is set to indicate Far-End Transmit mode, this register's contents will be transmitted as the BIST FIS 3rd DW regardless of whether or not the T bit is set in the BFCS register. |

## 50.2.36 Manufacturing ID (MFID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:19 or 20, F:0] + F8h

**Default:** 08000FB1h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved. |
| 27:24 | 0x8<br>RO/V | **Dot portion of Process ID (DPID):** Indicates the dot.<br>Process is reflected in bits [7:0].<br>This 8-bit value is received via the IOSF Sideband SetID message. |
| 23:16 | 0x0<br>RO/V | **Stepping Identifier (SID):** This field is incremented for each stepping of the part.<br>***Note:*** This field can be used by software to differentiate steppings when the Revision ID may not change.<br>This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message. |
| 15:8 | 0xf<br>RO/V | **Manufacturer Identifier (MID):** This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message. |
| 7:0 | 0xb1<br>RO/V | **Process/Dot Identifier (PID):** Indicates the Process.<br>Dot is reflected in bits [27:24].<br>This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message. |

## 50.3 Registers in Memory Space—MXTBA

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) MXTBA described in Section 50.2.7, "MSI-X Table Base Address (MXTBA)—Offset 10h" on page 1883.

### 50.3.1 MSI-X Table Entries n Message Lower Address (MXTEnMLA)

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MXTBA] + 0h + n*10h (where n<=7)

**MXTBA Type:** PCI Configuration Register (Size: 32 bits)
**MXTBA Reference:** [B:0, D:(19,20), F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **MSI-X message lower address (MXMLA):** Specifies the lower 32-bit of the DWORD-aligned address (Addr[31:02]) of the MSI-X Message. |
| 1:0 | 0x0<br>RO | **RSVD0:** Reserved. |

### 50.3.2 MSI-X Table Entries n Message Upper Address (MXTEnMUA)

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MXTBA] + 4h + n*10h (where n<=7)

**MXTBA Type:** PCI Configuration Register (Size: 32 bits)
**MXTBA Reference:** [B:0, D:(19,20), F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **MSI-X message upper 32-bit address (MXMUA):** Specifies the lower 32-bit of the DWORD-aligned address (Addr[31:02]) of the MSI-X Message. |

## 50.3.3 MSI-X Table Entries n Message Data (MXTEnMD)

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MXTBA] + 8h + n*10h (where n<=7)

**MXTBA Type:** PCI Configuration Register (Size: 32 bits)
**MXTBA Reference:** [B:0, D:(19,20), F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **MSI-X message Data (MXMD):** Specifies the 32-bit Data of the MSI-X Message. |

## 50.4 Registers in Memory Space—ABAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) ABAR described in Section 50.2.11, "AHCI Base Address (ABAR)—Offset 24h" on page 1886.

### 50.4.1 HBA Capabilities (GHC_CAP)—Offset 0h

This register indicates basic capabilities of the HBA to driver software. The RWO bits in this register are only cleared upon PLTRST#. This register is not reset by FLR.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FF36FF07h

**BAR and Offset:** [ABAR] + 0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW/O | **Supports 64-bit Addressing (S64A):** Indicates the S-ATA controller can access 64-bit data structures.<br>The 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry are read/write. |
| 30 | 0x1<br>RW/O | **Supports Native Command Queuing Acceleration (SCQA):** Indicates the SATA controller supports Serial-ATA Native Command Queuing.<br>The HBA will handle DMA Setup FISes in hardware, including support for auto-activate optimization through the FIS. |
| 29 | 0x1<br>RW/O | **Supports SNotification Register (SSNTF):** When set to 1, indicates that the HBA supports the PxSNTF (SNotification) register and its associated functionality.<br>When cleared to 0., the HBA does not support the PxSNTF (SNotification) register and its associated functionality. |
| 28 | 0x1<br>RW/O | **Supports Mechanical Presence Switch (SMPS):** When set to 1, the HBA supports mechanical presence switches on its ports for use in hot plug operations.<br>When cleared to 0, this function is not supported. This value is loaded by the BIOS prior to OS initialization. |
| 27 | 0x1<br>RW/O | **Supports Staggered Spin-up (SSS):** Indicates whether the S-ATA controller supports staggered spin-up on its ports, for use in balancing power spikes.<br>This value is loaded by platform BIOS prior to OS initialization. |
| 26 | 0x1<br>RW/O | **Supports Aggressive Link Power Management (SALP):** Indicates the S-ATA controller supports auto-generating link requests to the partial or slumber states when there are no commands to process.<br>When cleared to 0, software shall treat the PxCMD.ALPE and PxCMD.ASP bits as reserved. |
| 25 | 0x1<br>RW/O | **Supports Activity LED (SAL):** Indicates the S-ATA controller supports a single output pin (SATALED#) which indicates activity. |
| 24 | 0x1<br>RW/O | **Supports Command List Override (SCLO):** When set to 1, indicates that the HBA supports the PxCMD.CLO bit and it's associated function.<br>When cleared to 0., The HBA is not capable of clearing the BSY and DRQ bits in the Status register in order to issue a software reset if these bits are still set from a previous operation. |
| 23:20 | 0x3<br>RW/O | **Interface Speed Support (ISS):** Indicates the maximum speed the S-ATA controller can support on its ports.<br>These encodings match the system software programmable PxSCTL.DET.SPD field.<br>• 0000 = Reserved<br>• 0001 = Gen 1 (1.5 Gbps)<br>• 0010 = Gen 2 (3 Gbps)<br>• 0011 = Gen 3 (6 Gbps)<br>• 0100 - 1111 = Reserved.<br>***Note:*** If (FFSATA0p0, FFSATA0p1, FFSATA0p2, FFSATA0p3, FFSATA0p4 and FFSATA0p5) is 1, this field is RWO defaulting to 0010 and ignores software write value of 0011.<br>If either FFSATA0p0, FFSATA0p1, FFSATA0p2, FFSATA0p3, FFSATA0p4 or FFSATA0p5 is 0, this field is RWO defaulting to 0011. |
| 19 | 0x0<br>RO | **Supports Non-Zero DMA Offsets (SNZO):** Reserved as per AHCI 1.3. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x1<br>RO | **Supports AHCI mode only (SAM):** The SATA controller may optionally support AHCI access mechanism only.<br>A value of 1 indicates that the SATA controller does not implement a legacy, task-file based register interface. |
| 17 | 0x1<br>RO | **Supports Port Multiplier (SPM):** The SATA controller may optionally support command-based switching Port Multipliers. |
| 16 | 0x0<br>RO | **FIS-based Switching Supported (FBSS):** Not supported. |
| 15 | 0x1<br>RO | **PIO Multiple DRQ Block (PMD):** If set to 1, the HBA supports multiple DRQ block data transfers for the PIO command protocol. |
| 14 | 0x1<br>RW/O | **Slumber State Capable (SSC):** The SATA controller supports the slumber state. |
| 13 | 0x1<br>RW/O | **Partial State Capable (PSC):** The SATA controller supports the partial state. |
| 12:8 | 0x1f<br>RO | **Number of Command Slots (NCS):** 1Fh indicating support for 32 slots. |
| 7 | 0x0<br>RO | **Command Completion Coalescing Supported (CCCS):** When set to 1, indicates that the HBA supports command completion coalescing. |
| 6 | 0x0<br>RW/O/V | **Enclosure Management Supported (EMS):** When set to 1, indicates that the HBA supports enclosure management. |
| 5 | 0x0<br>RW/O | **Supports External SATA (SXS):** When set to 1, indicates that the HBA has one or more Serial ATA ports that has a signal only connector that is externally accessible. |
| 4:0 | 0x7<br>RO/V | **Number of Ports (NP):** 0's based value indicating the maximum number of ports supported. |

## 50.4.2 Global HBA Control (GHC)—Offset 4h

This register controls various global actions of the HBA.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 4h

**Default:** 80000000h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **AHCI Enable (AE):** When set, indicates that an AHCI driver is loaded and communication to the HBA shall be via AHCI mechanisms. |
| 30:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RO | **Reserved:** Reserved |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** This global bit enables interrupts from the HBA.<br>When cleared (reset default), all interrupt sources from all ports are disabled.<br>When set, interrupts are enabled. |
| 0 | 0x0<br>RW/1S | **HBA Reset (HR):** When set by SW, this bit causes an internal reset of the HBA.<br>All state machines that relate to data transfers and native command queuing will return to an idle condition, and all ports will be re-initialized via COMRESET. |

## 50.4.3 Interrupt Status Register (IS)—Offset 8h

This register indicates which of the ports within the controller have an interrupt pending and require service.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 8h

**Default:** 00000000h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 7 (IPS7):** If set, indicates that port 7 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 7 physically. |
| 6 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 6 (IPS6):** If set, indicates that port 6 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 6 physically. |
| 5 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 5 (IPS5):** If set, indicates that port 5 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 5 physically. |
| 4 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 4 (IPS4):** If set, indicates that port 4 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 4 physically. |
| 3 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 3 (IPS3):** If set, indicates that port 3 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 3 physically. |
| 2 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 2 (IPS2):** If set, indicates that port 2 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 2 physically. |
| 1 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 1 (IPS1):** If set, indicates that port 1 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 1 physically. |
| 0 | 0x0<br>RW/1C/V | **Interrupt Pending Status Port 0 (IPS0):** If set, indicates that port 0 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt.<br>This bit is only applicable to project(s): That has port 0 physically. |

## 50.4.4 Ports Implemented (GHC_PI)—Offset Ch

This register indicates which ports are exposed to the HBA. It indicates which ports that the device supports are available for software to use. Any available port may not be implemented. This register is not reset by FLR. There is BIOS programming requirement on the PI register. Please refer to section 7.9.13.1.1 for details.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + Ch

**Default:** 00000000h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7 | 0x0<br>RW/O/V | **Port 7 Implemented (PI7):** If set, then port 7 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 7 is not available. |
| 6 | 0x0<br>RW/O/V | **Port 6 Implemented (PI6):** If set, then port 6 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 6 is not available. |
| 5 | 0x0<br>RW/O/V | **Port 5 Implemented (PI5):** If set, then port 5 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 5 is not available. |
| 4 | 0x0<br>RW/O/V | **Port 4 Implemented (PI4):** If set, then port 4 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 4 is not available. |
| 3 | 0x0<br>RW/O/V | **Port 3 Implemented (PI3):** If set, then port 3 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 3 is not available. |
| 2 | 0x0<br>RW/O/V | **Port 2 Implemented (PI2):** If set, then port 2 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 2 is not available. |
| 1 | 0x0<br>RW/O/V | **Port 1 Implemented (PI1):** If set, then port 1 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 1 is not available. |
| 0 | 0x0<br>RW/O/V | **Port 0 Implemented (PI0):** If set, then port 0 is available for use.<br>If cleared, the port is not available for use.<br>This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 0 is not available. |

## 50.4.5 AHCI Version (VS)—Offset 10h

This register indicates the major and minor version of the AHCI specification. It is BCD encoded. The upper two bytes represent the major version number, and the lower two bytes represent the minor version number. Example: Version 3.12 would be represented as 00030102h.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 10h

**Default:** 00010300h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x1<br>RO | **Major Version Number (MJR):** Indicates the major version is 1 |
| 15:0 | 0x300<br>RO | **Minor Version Number (MNR):** Indicates the minor version is 30 |

## 50.4.6 Enclosure Management Control (EM_CTL)—Offset 20h

This register is used to control and obtain status for the enclosure management interface. The register includes information on the attributes of the implementation, enclosure management messages supported, the status of the interface, whether any messages are pending, and is used to initiate sending messages. This register is not implemented if enclosure management is not supported (i.e. CAP.EMS = 0).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 20h

**Default:** 07010000h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27 | 0x0<br>RO | **Port Multiplier Support (ATTR_PM):** The HBA does not support enclosure management messages for devices attached via a Port Multiplier.<br>Software should use the Serial ATA enclosure management bridge that is built into many Port Multipliers for enclosure services with these devices.<br>For more information on Serial ATA enclosure management bridges, refer to the Serial ATA II: Extensions to Serial ATA 1.0a revision 1.2 specification. |
| 26 | 0x1<br>RW/O | **Activity LED Hardware Driven (ATTR_ALHD):** If set to 1, the HBA drives the activity LED for the LED message type in hardware and does not utilize software settings for this LED.<br>The HBA does not begin transmitting the hardware based activity signal until after software has written CTL.TM=1 after a reset condition. |
| 25 | 0x1<br>RO | **Transmit Only (ATTR_XMT):** If set to 1, the HBA only supports transmitting messages and does not support receiving messages.<br>If cleared to 0, the HBA supports transmitting and receiving messages. |
| 24 | 0x1<br>RO | **Single Message Buffer (ATTR_SMB):** If set to 1, the HBA has one message buffer that is shared for messages to transmit and messages received.<br>In this case, unsolicited receive messages are not supported and it is software's responsibility to manage access to this buffer.<br>If cleared to 0, there are separate receive and transmit buffers such that unsolicited messages could be supported. |
| 23:20 | 0x0<br>RO | **RSVD1:** Reserved |
| 19 | 0x0<br>RO | **SGPIO Enclosure Management Messages (SUPP_SGPIO):** If set to 1, the HBA supports the SGPIO register interface message type. |
| 18 | 0x0<br>RO | **SES-2 Enclosure Management Messages (SUPP_SES2):** If set to 1, the HBA supports the SES-2 message type. |
| 17 | 0x0<br>RO | **SAF-TE Enclosure Management Messages (SUPP_SAFTE):** If set to 1, the HBA supports the SAF-TE message type. |
| 16 | 0x1<br>RO | **LED Message Types (SUPP_LED):** If set to 1, the HBA supports the LED message type defined in LED Message Type. |
| 15:10 | 0x0<br>RO | **RSVD2:** Reserved |
| 9 | 0x0<br>RW/1S | **Reset (RST):** When set to 1 by software, the HBA shall reset all enclosure management message logic and take all appropriate reset actions to ensure messages can be transmitted/received after the reset. After the HBA completes the reset operation, the HBA shall set the value to 0. A write of 0 by software to this field shall have no effect. |
| 8 | 0x0<br>RW/1S | **Transmit Message (CTL_TM):** When set to 1 by software, the HBA shall transmit the message contained in the message buffer.<br>When the message is completely sent, the HBA shall clear this bit to 0.<br>A write of 0 to this bit by software shall have no effect.<br>Software shall not change the contents of the message buffer while CTL.TM is set to 1. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RO | **RSVD3:** Reserved |
| 0 | 0x0<br>RO | **Message Received (STS_MR):** Message received is not supported. |

## 50.4.7 HBA Capabilities Extended (GHC_CAP2)—Offset 24h

This register indicates basic capabilities of the HBA to driver software. The RWO bits in this register are only cleared upon PLTRST#. This register is not reset by FLR.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 24h

**Default:** 0000003Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x1<br>RW/O | **DEVSLP Entrance from Slumber Only (DESO):** This field specifies that the HBA shall only assert DEVSLP if the interface is in Slumber.<br>When this bit is set to 1, the HBA shall ignore software directed entrance to DEVSLP via PxCMD.ICC unless PxSSTS.IPM = 6h. |
| 4 | 0x1<br>RW/O/V | **Supports Aggressive DEVSLP Management (SADM):** When set to 1, the HBA supports hardware assertion of the DEVSLP signal after the idle timeout expires. |
| 3 | 0x1<br>RW/O/V | **Supports DEVSLP (SDS):** When set to 1, the HBA supports the DEVSLP feature. When cleared to 0, DEVSLP is not supported. |
| 2 | 0x1<br>RW/O/V | **Automatic Partial to Slumber Transitions (APST):** When set to 1, the HBA supports Automatic Partial to Slumber Transitions. |
| 1 | 0x0<br>RO | **RSVD1:** Reserved |
| 0 | 0x0<br>RO | **Reserved:** Reserved |

## 50.4.8 Vendor Specific (VSP)—Offset A0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000048h

**BAR and Offset:** [ABAR] + A0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **RSVD0:** Reserved |
| 6 | 0x1<br>RO | **Software Feature Mask Supported (SFMS):** Set to 1 if the platform is enabled for premium storage features mask (SFM) as described in VS MMIO space at offset ABAR[RPID.(OFST*4)+4]. |
| 5 | 0x0<br>RO/V | **Premium Features Supported (PFS):** Set to 1 if the platform is enabled for premium storage features (PFB) as described in VS MMIO space at offset ABAR[RPID.OFST*4].<br>Set to 0 if the platform is not enabled for premium storage features. |
| 4 | 0x0<br>RO/V | **Platform Type (PT):** Set to 1 if mobile platform. Clear (0) if desktop. |
| 3 | 0x1<br>RO | **Supports RAID Platform ID Reporting (SRPIR):** If set to 1, then indicates that the RAID Platform ID is reported via ABAR + C0h.<br>Set to 0 if this ID is not reported via MMIO space. |
| 2:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 50.4.9 Vendor Specific Capabilities Register (VS_CAP)—Offset A4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 001002DEh

**BAR and Offset:** [ABAR] + A4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:16 | 0x10<br>RW/O | **NVM Remapped Register Offset (NRMO):** Specifies the offset (in 128B unit) within ABAR as to where the PCIe NAND memory BAR register space is remapped.<br>For example, NRMO=1 means ABAR + 128B.<br>This allows the remapped offset to shift between ABAR + 0B, and ABAR + 512KK - 128B, with 128B step.<br>The remapped offset into the AHCI memory space must not overlap with the memory space used for SATA.<br>The remapped offset into the AHCI memory space and the remapped size must not exceed the allocated AHCI memory space of the integrated AHCI controller.<br>Only valid when NRMBE = 1.<br>The reset default of this field is 10h, this places the start of the PCIe NAND memory BAR register space at ABAR + 2K. This field is not reset by FLR. |
| 15:13 | 0x0<br>RO | **RSVD1:** Reserved |
| 12:1 | 0x16f<br>RW/O | **Memory Space Limit. (MSL):** This field specifies the size (in 128B unit) of the remapped memory space for the PCIe NAND device.<br>It is a 0-based field. For example, MSL=1 means 256B.<br>This allows the remapped size to shift between 128B and 512K with the step of 128B.Memory BAR offset from 0 to MSL of the PCIe NAND device are remapped under the integrated AHCI controller memory space.<br>The remapped offset into the AHCI memory space and the value programmed in this field must not exceed the allocated AHCI memory space of the integrated AHCI controller.<br>Only valid when NRMBE = 1.<br>The reset default of this field is 16Fh which specifies the size of the remapped memory space as 34k.<br>This field is not reset by FLR. |
| 0 | 0x0<br>RW/O | **PCIe NAND Memory BAR Remapped Enable (NRMBE):** Set to 1 if a PCIe NAND device is present and remapping of its memory BAR register space is enabled.<br>Cleared to 0 if there is no PCIe NAND device present or the remapping of its memory BAR register space is disabled.<br>This bit is not reset by FLR. |

## 50.4.10 RAID Platform ID (RPID)—Offset C0h

This register is used by the Intel Matrix Storage Manager OROM to match the features supported by the OROM with the platform on which the OROM is executing. This prevents the use of an OROM designed for newer chipsets from being use on older chipsets as this could reduce up-sell potential.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + C0h

**Default:** 00311C02h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x31<br>RO | **Offset (OFST):** The offset of the Premium Feature Block (PFB) in DWords from the beginning of the ABAR. SFM follows directly after PFB. |
| 15:0 | 0x1c02<br>RO/V | **RAID Platform ID (RPID):** Specifies the DID value that has been assigned to the platform. This is the same DID that is reported by the SATA controller when SATAGC.AIE is set to 1 except that the DID is always reported through this register, regardless if the programming of SATAGC.AIE. |

## 50.4.11 SW Feature Mask (SFM)—Offset C8h

The following will be programmed by the BIOS when VS_CAP.SFMS == 1. The feature mask is used by SW to determine which non-premium features shall be supported by SW. These register bits are not reset by FLR since they are programmed by BIOS.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [ABAR] + C8h

**Default:** 003Fh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:10 | 0x0<br>RW/O | **OROM UI Normal Delay. (OROM_UI_Normal_Delay):** Values of these bits specify the delay of the OROM UI Splash Screen in a normal status.<br>• 00 = 2 secs (default and previous value)<br>• 01 = 4 secs<br>• 10 = 6 secs<br>• 11 = 8 secs.<br>If bit 5 == 0, then these values are disregarded. |
| 9 | 0x0<br>RW/O | **Smart Response Technology. (Smart_Response_Technology):** If set to '1', then Smart Response Technology is enabled. If cleared to '0', the feature is disabled. |
| 8 | 0x0<br>RW/O | **RRT Only on ESATA (IRRT_Only_on_ESATA):** If set to 1, then only RRT volumes can span internal and external SATA ports (e.g. eSATA).<br>If cleared to 0, then any RAID volume can span internal and external SATA ports (e.g. eSATA). |
| 7 | 0x0<br>RW/O | **LED Locate (LED_Locate):** If set to 1, then LED/SGPIO hardware is attached and the ping to locate feature is enabled in the OS. |
| 6 | 0x0<br>RW/O | **HDDUNLOCK:** If set to 1, then HDD password unlock is enabled in the OS. |
| 5 | 0x1<br>RW/O | **OROM UI and BANNER (OROM_UI_and_BANNER):** If set to 1, then the OROM UI is displayed. When cleared to 0, the OROM UI and BANNER are not displayed if all disks and volumes have a normal status. |
| 4 | 0x1<br>RW/O | **RRT (IRRT):** If set to 1, then Rapid Recovery Technology is enabled. |
| 3 | 0x1<br>RW/O | **R5:** If set to 1, then RAID5 is enabled |
| 2 | 0x1<br>RW/O | **R10:** If set to 1, then RAID10 is enabled |
| 1 | 0x1<br>RW/O | **R1:** If set to 1, then RAID1 is enabled |
| 0 | 0x1<br>RW/O | **R0:** If set to 1, then RAID0 is enabled |

## 50.4.12 Port [0-7] Command List Base Address (PxCLB0)—Offset 100h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 100h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.13 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU0)—Offset 104h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 104h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.14 Port [0-7] FIS Base Address (PxFB0)—Offset 108h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 108h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.15 Port [0-7] FIS Base Address Upper 32-bits (PxFBU0)—Offset 10Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 10Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.16 Port [0-7] Interrupt Status (PxIS0)—Offset 110h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 110h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.17 Port [0-7] Interrupt Enable (PxIE0)—Offset 114h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 114h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'.<br>The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.18 Port [0-7] Command (PxCMD0)—Offset 118h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 118h

**Default:** 00000004h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleared to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running.<br>Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list.<br>When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.19 Port [0-7] Serial ATA Status (PxSSTS0)—Offset 128h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 128h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.20 Port [0-7] Serial ATA Control (PxSCTL0)—Offset 12Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 12Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.21 Port [0-7] Serial ATA Error (PxSERR0)—Offset 130h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 130h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0<br>RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.22 Port [0-7] Serial ATA Active (PxSACT0)—Offset 134h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 134h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.23 Port [0-7] Commands Issued (PxCI0)—Offset 138h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 138h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.24 Port [0-7] SNotification (PxSNTF0)—Offset 13Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 13Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.25 Port [0-7] Command List Base Address (PxCLB1)—Offset 180h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 180h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.26 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU1)—Offset 184h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 184h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.27 Port [0-7] FIS Base Address (PxFB1)—Offset 188h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 188h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.28 Port [0-7] FIS Base Address Upper 32-bits (PxFBU1)—Offset 18Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 18Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.29 Port [0-7] Interrupt Status (PxIS1)—Offset 190h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 190h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X.<br>This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.30 Port [0-7] Interrupt Enable (PxIE1)—Offset 194h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 194h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt.<br>BIOS is required to program this field to '0'.<br>The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.31 Port [0-7] Command (PxCMD1)—Offset 198h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 198h

**Default:** 00000004h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleared to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only. If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RW/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RW/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running.<br>Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0. After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list.<br>When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.32 Port [0-7] Serial ATA Status (PxSSTS1)—Offset 1A8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1A8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.33 Port [0-7] Serial ATA Control (PxSCTL1)—Offset 1ACh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1ACh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.34 Port [0-7] Serial ATA Error (PxSERR1)—Offset 1B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1B0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.35 Port [0-7] Serial ATA Active (PxSACT1)—Offset 1B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1B4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.36 Port [0-7] Commands Issued (PxCI1)—Offset 1B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1B8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.37 Port [0-7] SNotification (PxSNTF1)—Offset 1BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 1BCh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST. |

## 50.4.38 Port [0-7] Command List Base Address (PxCLB2)—Offset 200h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 200h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.39 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU2)—Offset 204h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 204h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.40 Port [0-7] FIS Base Address (PxFB2)—Offset 208h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 208h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.41 Port [0-7] FIS Base Address Upper 32-bits (PxFBU2)—Offset 20Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 20Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.42 Port [0-7] Interrupt Status (PxIS2)—Offset 210h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 210h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X.<br>This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit.<br>PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.43 Port [0-7] Interrupt Enable (PxIE2)—Offset 214h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 214h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'.<br>The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.44 Port [0-7] Command (PxCMD2)—Offset 218h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000004h

**BAR and Offset:** [ABAR] + 218h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleard to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved. BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running.<br>Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list.<br>When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state.<br>PxTFD shall be updated also. |

## 50.4.45 Port [0-7] Serial ATA Status (PxSSTS2)—Offset 228h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 228h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.46 Port [0-7] Serial ATA Control (PxSCTL2)—Offset 22Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 22Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.47 Port [0-7] Serial ATA Error (PxSERR2)—Offset 230h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 230h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.48 Port [0-7] Serial ATA Active (PxSACT2)—Offset 234h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 234h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.49 Port [0-7] Commands Issued (PxCI2)—Offset 238h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 238h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.50 Port [0-7] SNotification (PxSNTF2)—Offset 23Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 23Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.51 Port [0-7] Command List Base Address (PxCLB3)—Offset 280h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 280h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.52 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU3)—Offset 284h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 284h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port. This base is used when fetching commands to execute. Note that these bits are not reset on a HBA reset. |

## 50.4.53 Port [0-7] FIS Base Address (PxFB3)—Offset 288h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 288h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.54 Port [0-7] FIS Base Address Upper 32-bits (PxFBU3)—Offset 28Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 28Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.55 Port [0-7] Interrupt Status (PxIS3)—Offset 290h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 290h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>1=Change in Current Connect Status.<br>0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X.<br>This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>***Note:*** This bit does not directly reflect the PxSERR.DIAG.F bit.<br>PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.56 Port [0-7] Interrupt Enable (PxIE3)—Offset 294h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 294h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'.<br>The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.57 Port [0-7] Command (PxCMD3)—Offset 298h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [ABAR] + 298h

**Default:** 00000004h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleared to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running. Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list.<br>When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.58 Port [0-7] Serial ATA Status (PxSSTS3)—Offset 2A8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2A8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.59 Port [0-7] Serial ATA Control (PxSCTL3)—Offset 2ACh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2ACh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.60 Port [0-7] Serial ATA Error (PxSERR3)—Offset 2B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2B0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.61 Port [0-7] Serial ATA Active (PxSACT3)—Offset 2B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2B4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.62 Port [0-7] Commands Issued (PxCI3)—Offset 2B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2B8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.63 Port [0-7] SNotification (PxSNTF3)—Offset 2BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 2BCh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.64 Port [0-7] Command List Base Address (PxCLB4)—Offset 300h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 300h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>Note that these bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.65 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU4)—Offset 304h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 304h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.66 Port [0-7] FIS Base Address (PxFB4)—Offset 308h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 308h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.67 Port [0-7] FIS Base Address Upper 32-bits (PxFBU4)—Offset 30Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 30Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.68 Port [0-7] Interrupt Status (PxIS4)—Offset 310h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 310h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>***Note:*** This bit does not directly reflect the PxSERR.DIAG.F bit.<br>PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.69 Port [0-7] Interrupt Enable (PxIE4)—Offset 314h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 314h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt.<br>BIOS is required to program this field to '0'. The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.70 Port [0-7] Command (PxCMD4)—Offset 318h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000004h

**BAR and Offset:** [ABAR] + 318h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleard to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0 when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running. Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list. When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.71 Port [0-7] Serial ATA Status (PxSSTS4)—Offset 328h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 328h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.72 Port [0-7] Serial ATA Control (PxSCTL4)—Offset 32Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 32Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.73 Port [0-7] Serial ATA Error (PxSERR4)—Offset 330h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 330h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.74 Port [0-7] Serial ATA Active (PxSACT4)—Offset 334h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 334h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.75 Port [0-7] Commands Issued (PxCI4)—Offset 338h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 338h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.76 Port [0-7] SNotification (PxSNTF4)—Offset 33Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 33Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions. Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.77 Port [0-7] Command List Base Address (PxCLB5)—Offset 380h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 380h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.78 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU5)—Offset 384h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 384h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.79 Port [0-7] FIS Base Address (PxFB5)—Offset 388h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 388h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.80 Port [0-7] FIS Base Address Upper 32-bits (PxFBU5)—Offset 38Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 38Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.81 Port [0-7] Interrupt Status (PxIS5)—Offset 390h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 390h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.82 Port [0-7] Interrupt Enable (PxIE5)—Offset 394h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 394h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'.<br>The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.83 Port [0-7] Serial ATA Status (PxSSTS5)—Offset 3A8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3A8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.84 Port [0-7] Serial ATA Control (PxSCTL5)—Offset 3ACh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3ACh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.85 Port [0-7] Serial ATA Error (PxSERR5)—Offset 3B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3B0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.86 Port [0-7] Serial ATA Active (PxSACT5)—Offset 3B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3B4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.87 Port [0-7] Commands Issued (PxCI5)—Offset 3B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3B8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.88 Port [0-7] SNotification (PxSNTF5)—Offset 3BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 3BCh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0. PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>***Note:*** While this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.89 Port [0-7] Device Sleep (PxDEVSLP5)—Offset 3C4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 1E022852h

**BAR and Offset:** [ABAR] + 3C4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:25 | 0xf<br>RW/O/V | **DITO Multiplier (DM):** 0's based value that specifies the DITO multiplier that the HBA applies to the specified DITO value, effectively extending the range of DITO from 1ms to 16368ms.<br>A value of 0h indicates a multiplier of 1.<br>A maximum multiplier of 16 may be applied. |
| 24:15 | 0x4<br>RW/V | **DEVSLP Idle Timeout (DITO):** This field specifies the amount of the time (in approximate 1ms granularity) that the HBA shall wait before driving the DEVSLP signal. |
| 14:10 | 0xa<br>RW/V | **DEVSLP Minimum Assertion Time (MDAT):** This field specifies the minimum amount of time (in 1ms granularity) that the HBA must assert the DEVSLP signal before it may be de-asserted.<br>The nominal value is 10ms and the minimum is 1ms depending on device identification information. |
| 9:2 | 0x14<br>RW/V | **DEVSLP Exit Timeout (DETO):** This field specifies the maximum duration (in approximate 1ms granularity) from DEVSLP de-assertion until the device is ready to accept OOB.<br>The nominal value is 20ms while the max value is 255ms depending on device identification information. |
| 1 | 0x1<br>RW/O | **DEVSLP Present (DSP):** If set to '1', the platform supports DEVSLP on this port.<br>If cleared to '0', the platform does not support DEVSLP on this port. |
| 0 | 0x0<br>RW/V | **Aggressive DEVSLP Enable (ADSE):** This bit is read/write for HBAs that support aggressive DEVSLP management (CAP2. SADM = 1). |

## 50.4.90 Port [0-7] Command List Base Address (PxCLB6)—Offset 400h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 400h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.91 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU6)—Offset 404h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 404h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute. Note that these bits are not reset on a HBA reset. |

## 50.4.92 Port [0-7] FIS Base Address (PxFB6)—Offset 408h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 408h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.93 Port [0-7] FIS Base Address Upper 32-bits (PxFBU6)—Offset 40Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 40Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.94 Port [0-7] Interrupt Status (PxIS6)—Offset 410h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 410h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X.<br>This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.95 Port [0-7] Interrupt Enable (PxIE6)—Offset 414h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 414h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'. The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.96 Port [0-7] Command (PxCMD6)—Offset 418h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000004h

**BAR and Offset:** [ABAR] + 418h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleared to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):**<br>If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>***Note:*** This bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running.<br>Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1.<br>For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU. When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS. This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list.<br>When cleared, the HBA may not process the command list.<br>Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 ot a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.97 Port [0-7] Task File Data (PxTFD6)—Offset 420h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0000007Fh

**BAR and Offset:** [ABAR] + 420h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:8 | 0x0<br>RO/V | **Error (ERR):** Contains the latest copy of the task file error register. |
| 7 | 0x0<br>RO/V | **Status Busy (STS_BSY):** Status - Indicates the interface is busy. |
| 6:4 | 0x7<br>RO/V | **Status Rsvd0 (STS_RSVD0):** Status - Not Applicable. |
| 3 | 0x1<br>RO/V | **Status Drq (STS_DRQ):** Status - Indicates a data transfer is requested. |
| 2:1 | 0x3<br>RO/V | **Status Rsvd1 (STS_RSVD1):** Status - Not Applicable. |
| 0 | 0x1<br>RO/V | **Status Err (STS_ERR):** Status - Indicates an error during the transfer. |

## 50.4.98 Port [0-7] Signature (PxSIG6)—Offset 424h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FFFFFFFFh

**BAR and Offset:** [ABAR] + 424h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO/V | **Signature (SIG):** Contains the signature received from a device on the first D2H Register FIS. |

## 50.4.99 Port [0-7] Serial ATA Status (PxSSTS6)—Offset 428h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 428h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.100 Port [0-7] Serial ATA Control (PxSCTL6)—Offset 42Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 42Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.101 Port [0-7] Serial ATA Error (PxSERR6)—Offset 430h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 430h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.102 Port [0-7] Serial ATA Active (PxSACT6)—Offset 434h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 434h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.103 Port [0-7] Commands Issued (PxCI6)—Offset 438h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 438h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.104 Port [0-7] SNotification (PxSNTF6)—Offset 43Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 43Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0.<br>PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>***Note:*** While this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.105 Port [0-7] Device Sleep (PxDEVSLP6)—Offset 444h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 1E022852h

**BAR and Offset:** [ABAR] + 444h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28:25 | 0xf<br>RW/O/V | **DITO Multiplier (DM):** 0's based value that specifies the DITO multiplier that the HBA applies to the specified DITO value, effectively extending the range of DITO from 1ms to 16368ms.<br>A value of 0h indicates a multiplier of 1. A maximum multiplier of 16 may be applied. |
| 24:15 | 0x4<br>RW/V | **DEVSLP Idle Timeout (DITO):** This field specifies the amount of the time (in approximate 1ms granularity) that the HBA shall wait before driving the DEVSLP signal.<br>Hardware reloads its port specific DEVSLP timer with this value each time the port transitions out of DEVSLP state. |
| 14:10 | 0xa<br>RW/V | **DEVSLP Minimum Assertion Time (MDAT):** This field specifies the minimum amount of time (in 1ms granularity) that the HBA must assert the DEVSLP signal before it may be de-asserted.<br>The nominal value is 10ms and the minimum is 1ms depending on device identification information. |
| 9:2 | 0x14<br>RW/V | **DEVSLP Exit Timeout (DETO):** This field specifies the maximum duration (in approximate 1ms granularity) from DEVSLP de-assertion until the device is ready to accept OOB.<br>The nominal value is 20ms while the max value is 255ms depending on device identification information. |
| 1 | 0x1<br>RW/O | **DEVSLP Present (DSP):** If set to '1', the platform supports DEVSLP on this port.<br>If cleared to '0', the platform does not support DEVSLP on this port. |
| 0 | 0x0<br>RW/V | **Aggressive DEVSLP Enable (ADSE):** This bit is read/write for HBAs that support aggressive DEVSLP management (CAP2. SADM = 1). |

## 50.4.106 Port [0-7] Command List Base Address (PxCLB7)—Offset 480h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 480h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RW | **Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute.<br>This address must be 1K aligned as indicated by bits 31:10 being read/write.<br>***Note:*** These bits are not reset on a HBA reset. |
| 9:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.107 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU7)—Offset 484h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 484h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port.<br>This base is used when fetching commands to execute.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.108 Port [0-7] FIS Base Address (PxFB7)—Offset 488h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 488h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **FIS Base Address (FB):** Indicates the 32-bit base for received FISes.<br>This address must be 256-byte aligned as indicated by bits 31:08 being read/write.<br>When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned.<br>***Note:*** These bits are not reset on a HBA reset. |
| 7:0 | 0x0<br>RO | **RSVD0:** Reserved |

## 50.4.109 Port [0-7] FIS Base Address Upper 32-bits (PxFBU7)—Offset 48Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 48Ch

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port.<br>***Note:*** These bits are not reset on a HBA reset. |

## 50.4.110 Port [0-7] Interrupt Status (PxIS7)—Offset 490h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 490h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW/1C/V | **Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set. |
| 29 | 0x0<br>RW/1C/V | **Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer.<br>In PCI, such an indication would be a target or master abort. |
| 28 | 0x0<br>RW/1C/V | **Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory. |
| 27 | 0x0<br>RW/1C/V | **Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop. |
| 26 | 0x0<br>RW/1C/V | **Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW/1C/V | **Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command. |
| 23 | 0x0<br>RW/1C/V | **Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected.<br>The IPMS bit may be set during enumeration process.<br>It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier. |
| 22 | 0x0<br>RO/V | **PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state.<br>This bit reflects the state of PxSERR.DIAG.N.<br>This bit is RO and is only cleared when PxSERR.DIAG.N is cleared. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/1C/V | **Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device.<br>This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set). |
| 6 | 0x0<br>RO/V | **Port Connect Change Status (PCS):**<br>• 1=Change in Current Connect Status.<br>• 0=No change in Current Connect Status.<br>This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared. |
| 5 | 0x0<br>RW/1C/V | **Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data. |
| 4 | 0x0<br>RO/V | **Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory.<br>This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0.<br>Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory.<br>Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync. |
| 3 | 0x0<br>RW/1C/V | **Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C/V | **DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory. |
| 1 | 0x0<br>RW/1C/V | **PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred.<br>This bit shall be set even if the data transfer resulted in an error. |
| 0 | 0x0<br>RW/1C/V | **Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory. |

## 50.4.111 Port [0-7] Interrupt Enable (PxIE7)—Offset 494h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 494h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect. |
| 30 | 0x0<br>RW | **Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt. |
| 29 | 0x0<br>RW | **Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt. |
| 28 | 0x0<br>RW | **Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt. |
| 27 | 0x0<br>RW | **Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt. |
| 26 | 0x0<br>RW | **Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt. |
| 25 | 0x0<br>RO | **RSVD0:** Reserved |
| 24 | 0x0<br>RW | **Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt. |
| 23 | 0x0<br>RW | **Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt.<br>BIOS is required to program this field to '0'. The same applies to AHCI driver. |
| 22 | 0x0<br>RW | **PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt. |
| 21:8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW/V | **Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt. |
| 6 | 0x0<br>RW | **Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt. |
| 5 | 0x0<br>RW | **Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt. |
| 4 | 0x0<br>RW | **Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt. |
| 3 | 0x0<br>RW | **Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt. |
| 2 | 0x0<br>RW | **DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt. |
| 1 | 0x0<br>RW | **PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt. |
| 0 | 0x0<br>RW | **Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt. |

## 50.4.112 Port [0-7] Command (PxCMD7)—Offset 498h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000004h

**BAR and Offset:** [ABAR] + 498h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Interface Communication Control (ICC):** This is a four bit field which can be used to control reset and power states of the interface.<br>If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested.<br>If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect. |
| 27 | 0x0<br>RW | **Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h.<br>If CAP.SALP is cleard to 0., software shall treat this bit as reserved. |
| 26 | 0x0<br>RW | **Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit.<br>Software shall only set this bit to 1 if CAP.SALP is set to 1.<br>If CAP.SALP is cleared to 0, software shall treat this bit as reserved.<br>BIOS is recommended to program this field to 1. |
| 25 | 0x0<br>RW | **Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI.<br>When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.<br>This bit is set by software. |
| 24 | 0x0<br>RW | **Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device.<br>This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active. |
| 23 | 0x0<br>RW | **Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions.<br>When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions.<br>Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved. |
| 22 | 0x0<br>RO | **FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching. |
| 21 | 0x0<br>RW/O | **External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device.<br>When set to 1, CAP.SXS must also be set to 1.<br>When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only.<br>If ESP is set to '1', then the port may experience hot plug events. |
| 20 | 0x0<br>RO | **Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect. |
| 19 | 0x0<br>RW/O | **Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port.<br>If cleared to 0, the platform does not support a mechanical presence switch attached to this port. |
| 18 | 0x0<br>RW/O | **Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged.<br>SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example).<br>This bit can be used by system software to indicate a feature such as eject device to the end-user. |
| 17 | 0x0<br>RW/V | **Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port.<br>When cleared, a Port Multiplier is not attached to the HBA for this port.<br>This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.<br>Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15 | 0x0<br>RO/V | **Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running. |
| 14 | 0x0<br>RO/V | **FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running.<br>Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any |
| 13 | 0x0<br>RO/V | **Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0.<br>If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1.<br>If CAP.SMPS is set to 0 then this bit is cleared to 0.<br>Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1. |
| 12:8 | 0x0<br>RO | **Current Command Slot (CCS):** Indicates the current command slot the HBA is processing.<br>This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA.<br>This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command.<br>When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0.<br>After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0.<br>After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1. For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1. |
| 7:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RW | **FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU.<br>When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence. |
| 3 | 0x0<br>RW/1S | **Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0.<br>This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register.<br>The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0.<br>A write to this register with a value of 0 shall have no effect. |
| 2 | 0x1<br>RO | **Power On Device (POD):** The SATA controller does not support cold presence detect. |
| 1 | 0x0<br>RW/V | **Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS.<br>This bit is read only 1. for HBAs that do not support staggered spin-up.<br>On an edge detect from 0. to 1, the HBA shall start a COMRESET initialization sequence to the device.<br>Clearing this bit causes no action on the interface.<br>Clearing this bit to0 does not cause any OOB signal to be sent on the interface.<br>When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode. |
| 0 | 0x0<br>RW | **Start (ST):** When set, the HBA may process the command list. When cleared, the HBA may not process the command list. Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0.<br>Whenever this bit is changed from a 1 to a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also. |

## 50.4.113 Port [0-7] Task File Data (PxTFD7)—Offset 4A0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0000007Fh

**BAR and Offset:** [ABAR] + 4A0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:8 | 0x0<br>RO/V | **Error (ERR):** Contains the latest copy of the task file error register. |
| 7 | 0x0<br>RO/V | **Status Busy (STS_BSY):** Status - Indicates the interface is busy. |
| 6:4 | 0x7<br>RO/V | **Status Rsvd0 (STS_RSVD0):** Status - Not Applicable. |
| 3 | 0x1<br>RO/V | **Status Drq (STS_DRQ):** Status - Indicates a data transfer is requested. |
| 2:1 | 0x3<br>RO/V | **Status Rsvd1 (STS_RSVD1):** Status - Not Applicable. |
| 0 | 0x1<br>RO/V | **Status Err (STS_ERR):** Status - Indicates an error during the transfer. |

## 50.4.114 Port [0-7] Signature (PxSIG7)—Offset 4A4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FFFFFFFFh

**BAR and Offset:** [ABAR] + 4A4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO/V | **Signature (SIG):** Contains the signature received from a device on the first D2H Register FIS. |

## 50.4.115 Port [0-7] Serial ATA Status (PxSSTS7)—Offset 4A8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4A8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD0:** Reserved |
| 11:8 | 0x0<br>RO/V | **Interface Power Management (IPM):** Indicates the current interface state |
| 7:4 | 0x0<br>RO/V | **Current Interface Speed (SPD):** Indicates the negotiated interface communication speed. |
| 3:0 | 0x0<br>RO/V | **Device Detection (DET):** Indicates the interface device detection and Phy state. |

## 50.4.116 Port [0-7] Serial ATA Control (PxSCTL7)—Offset 4ACh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4ACh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RSVD0:** Reserved |
| 19:16 | 0x0<br>RW | **Port Multiplier Port (PMP):** This field is not used by AHCI. |
| 15:12 | 0x0<br>RW | **Select Power Management (SPM):** This field is not used by AHCI. |
| 11:8 | 0x0<br>RW | **Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to.<br>If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state. |
| 7:4 | 0x0<br>RW | **Speed Allowed (SPD):** Indicates the highest allowable speed of the interface. |
| 3:0 | 0x0<br>RW | **Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization. |

## 50.4.117 Port [0-7] Serial ATA Error (PxSERR7)—Offset 4B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4B0h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0 RW/1C/V | **Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes. |
| 15:0 | 0x0 RW/1C/V | **Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition.<br>If one or more of bits 11:8 of this register are set, the controller will stop the current transfer. |

## 50.4.118 Port [0-7] Serial ATA Active (PxSACT7)—Offset 4B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4B4h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry.<br>This field is cleared via the Set Device Bits FIS. |

## 50.4.119 Port [0-7] Commands Issued (PxCI7)—Offset 4B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4B8h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/1S/V | **Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device.<br>When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot.<br>Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1. |

## 50.4.120 Port [0-7] SNotification (PxSNTF7)—Offset 4BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 4BCh

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **RSVD0:** Reserved |
| 15:0 | 0x0<br>RW/1C/V | **PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set.<br>PM Port 0h sets bit 0. PM Port Fh sets bit 15.<br>Individual bits are cleared by software writing 1's to the corresponding bit positions.<br>***Note:*** While this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST |

## 50.4.121 Enclosure Management Message Format (EM_MF)—Offset 580h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 580h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RSVD0:** Reserved |
| 27:24 | 0x0<br>RW | **Message Type (MTYPE):** Specifies the type of the message.<br>The message types are:<br>• 0h = LED<br>• 1h = SAF-TE<br>• 2h = SES-2<br>• 3h = SGPIO (register based interface)<br>• All other values reserved. |
| 23:16 | 0x0<br>RW | **Data Size (DSIZE):** Specifies the data size in bytes.<br>If the message (enclosure services command) has a data buffer that is associated with it that is transferred, the size of that data buffer is specified in this field.<br>If there is no separate data buffer, this field shall have a value of '0'.<br>The data directly follows the message in the message buffer. This value should always be '0'. |
| 15:8 | 0x0<br>RW | **Message Size (MSIZE):** Specifies the size of the message in bytes.<br>The message size does not include the one Dword header.<br>A value of '0'. is invalid. The message size is always 4 bytes. |
| 7:0 | 0x0<br>RO | **RSVD1:** Reserved |

## 50.4.122 Enclosure Management LED (EM_LED)—Offset 584h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABAR] + 584h

**ABAR Type:** PCI Configuration Register (Size: 32 bits)
**ABAR Reference:** [B:0, D:(19,20), F:0] + 24h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Value (VAL):** This field describes the state of each LED for a particular location. There are three LEDs that may be supported by the HBA. Each LED has 3 bits of control.<br>LED values are:<br>• 000b - LED shall be off<br>• 001b - LED shall be solid on as perceived by human eye<br>• All other values reserved.<br>The LED bit locations are:<br>• Bits 2:0 - Activity LED (may be driven by hardware)<br>• Bits 5:3 - Vendor Specific LED (e.g. locate)<br>• Bits 8:6 - Vendor Specific LED (e.g. fault)<br>• Bits 15:9 - Reserved.<br>Vendor specific message is: Bit 3:0 - Vendor Specific Pattern; Bit 15:4 - Reserved. |
| 15:8 | 0x0<br>RW | **Port Multiplier Information (PM):** Specifies slot specific information related to Port Multiplier. Bits 3:0 specify the Port Multiplier port number for the slot that requires the status update.<br>• If a Port Multiplier is not attached to the device in the affected slot, the Port Multiplier port number shall be '0'.<br>• Bits 7:4 are reserved. SATA does not support LED messages for devices behind a Port Multiplier. This byte should be 0. |
| 7:0 | 0x0<br>RW | **HBA Information (HBA):** Specifies slot specific information related to the HBA.<br>• Bits 4:0 - HBA port number for the slot that requires the status update.<br>• Bit 5 - If set to '1', Value is a vendor specific message that applies to the entire enclosure. If cleared to '0', Value applies to the port specified in bits 4:0.<br>• Bits 7:6 - Reserved. |

§ §

# 51 USB Combo Controller - B0, D21, F0

## 51.1 Introduction and Index

The host-accessible registers for the USB Combo Controller are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 51.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 21 (decimal), Function 0. The offset addresses are listed.

**Table 51-1. Summary of PCI Configuration Registers—0/21/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "Vendor ID (VID)—Offset 0h" on page 2006 |
| 2 | 2 | 19D0 | "Device ID (DID)—Offset 2h" on page 2006 |
| 4 | 2 | 0000 | "Command (CMD)—Offset 4h" on page 2007 |
| 6 | 2 | 0290 | "Device Status (STS)—Offset 6h" on page 2008 |
| 8 | 1 | 00 | "Revision ID (RID)—Offset 8h" on page 2010 |
| 9 | 1 | 30 | "Programming Interface (PI)—Offset 9h" on page 2010 |
| A | 1 | 03 | "Sub Class Code (SCC)—Offset Ah" on page 2010 |
| B | 2 | 0C | "Base Class Code (BCC)—Offset Bh" on page 2010 |
| D | 1 | 00 | "Master Latency Timer (MLT)—Offset Dh" on page 2011 |
| E | 2 | 00 | "Header Type (HT)—Offset Eh" on page 2011 |
| 10 | 8 | 0000000000000004 | "Memory Base Address (MBAR)—Offset 10h" on page 2012 |
| 2C | 2 | 0000 | "USB Subsystem Vendor ID (SSVID)—Offset 2Ch" on page 2012 |
| 2E | 2 | 0000 | "USB Subsystem ID (SSID)—Offset 2Eh" on page 2013 |
| 34 | 1 | 70 | "Capabilities Pointer (CAP_PTR)—Offset 34h" on page 2013 |
| 3C | 1 | 00 | "Interrupt Line (ILINE)—Offset 3Ch" on page 2013 |
| 3D | 3 | 00 | "Interrupt Pin (IPIN)—Offset 3Dh" on page 2014 |
| 40 | 4 | 000001FD | "XHC System Bus Configuration 1 (XHCC1)—Offset 40h" on page 2014 |
| 44 | 4 | 0003C000 | "XHC System Bus Configuration 2 (XHCC2)—Offset 44h" on page 2016 |
| 50 | 4 | 00000000 | "Clock Gating (XHCLKGTEN)—Offset 50h" on page 2018 |
| 58 | 4 | 00000000 | "Audio Time Synchronization (AUDSYNC)—Offset 58h" on page 2021 |
| 60 | 1 | 30 | "Serial Bus Release Number (SBRN)—Offset 60h" on page 2021 |
| 61 | 1 | 60 | "Frame Length Adjustment (FLADJ)—Offset 61h" on page 2022 |
| 62 | 1 | 00 | "Best Effort Service Latency (BESL)—Offset 62h" on page 2023 |
| 70 | 1 | 01 | "PCI Power Management Capability ID (PM_CID)—Offset 70h" on page 2023 |
| 71 | 1 | 80 | "Next Item Pointer #1 (PM_NEXT)—Offset 71h" on page 2023 |
| 72 | 2 | C1C2 | "Power Management Capabilities (PM_CAP)—Offset 72h" on page 2024 |
| 74 | 2 | 0008 | "Power Management Control/Status (PM_CS)—Offset 74h" on page 2025 |
| 80 | 1 | 05 | "Message Signaled Interrupt CID (MSI_CID)—Offset 80h" on page 2026 |
| 81 | 1 | 00 | "Next item pointer (MSI_NEXT)—Offset 81h" on page 2026 |
| 82 | 2 | 0086 | "Message Signaled Interrupt Message Control (MSI_MCTL)—Offset 82h" on page 2027 |
| 84 | 4 | 00000000 | "Message Signaled Interrupt Message Address (MSI_MAD)—Offset 84h" on page 2028 |
| 88 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (MSI_MUAD)—Offset 88h" on page 2028 |
| 8C | 2 | 0000 | "Message Signaled Interrupt Message Data (MSI_MD)—Offset 8Ch" on page 2028 |
| 90 | 4 | F0140009 | "Device Idle Capability (DEVIDLE)—Offset 90h" on page 2029 |

## Table 51-1. Summary of PCI Configuration Registers—0/21/0 (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 94 | 4 | 01400010 | "Vendor Specific Header (VSHDR)—Offset 94h" on page 2029 |
| 98 | 4 | 00000000 | "SW LTR POINTER (SWLTRPTR)—Offset 98h" on page 2030 |
| 9C | 4 | 00080AC1 | "Device Idle Pointer Register (DEVIDLEPTR)—Offset 9Ch" on page 2030 |
| A0 | 2 | 0800 | "Device Idle Power ON Latency (DEVIDLEPOL)—Offset A0h" on page 2031 |
| A4 | 4 | 00002000 | "High Speed Configuration 2 (HSCFG2)—Offset A4h" on page 2032 |
| B0 | 4 | 00000000 | "XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1)—Offset B0h" on page 2033 |
| D0 | 4 | 00000000 | "XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1)—Offset D0h" on page 2033 |
| FC | 4 | 00000002 | "XHCC3—Offset FCh" on page 2034 |

### 51.1.2 Host Memory Space—MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 51-2. Summary of Memory Mapped I/O Registers—MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 80 | "Capability Registers Length (CAPLENGTH)—Offset 0h" on page 2035 |
| 2 | 2 | 0100 | "Host Controller Interface Version Number (HCIVERSION)—Offset 2h" on page 2035 |
| 4 | 4 | 08000820 | "Structural Parameters 1 (HCSPARAMS1)—Offset 4h" on page 2036 |
| 8 | 4 | 94000054 | "Structural Parameters 2 (HCSPARAMS2)—Offset 8h" on page 2036 |
| C | 4 | 00040001 | "Structural Parameters 3 (HCSPARAMS3)—Offset Ch" on page 2037 |
| 10 | 4 | 200077C1 | "Capability Parameters (HCCPARAMS)—Offset 10h" on page 2038 |
| 14 | 4 | 00003000 | "Doorbell Offset (DBOFF)—Offset 14h" on page 2039 |
| 18 | 4 | 00002000 | "Runtime Register Space Offset (RTSOFF)—Offset 18h" on page 2039 |
| 80 | 4 | 00000000 | "USB Command (USBCMD)—Offset 80h" on page 2040 |
| 84 | 4 | 00000001 | "USB Status (USBSTS)—Offset 84h" on page 2041 |
| 88 | 4 | 00000001 | "Page Size (PAGESIZE)—Offset 88h" on page 2042 |
| 94 | 4 | 00000000 | "Device Notification Control (DNCTRL)—Offset 94h" on page 2042 |
| 98 | 4 | 00000000 | "Command Ring Low (CRCR_LO)—Offset 98h" on page 2043 |
| 9C | 4 | 00000000 | "Command Ring High (CRCR_HI)—Offset 9Ch" on page 2043 |
| B0 | 4 | 00000000 | "Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h" on page 2044 |
| B4 | 4 | 00000000 | "Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h" on page 2044 |
| B8 | 4 | 00000000 | "Configure (CONFIG)—Offset B8h" on page 2044 |
| 480 | 4 | 000002A0 | "Port Status and Control USB2 (PORTSC1)—Offset 480h" on page 2045 |
| 484 | 4 | 00000000 | "Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h" on page 2047 |
| 48C | 4 | 00000000 | "Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch" on page 2048 |
| 490 | 4 | 000002A0 | "Port Status and Control USB2 (PORTSC2)—Offset 490h" on page 2049 |
| 494 | 4 | 00000000 | "Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h" on page 2051 |
| 49C | 4 | 00000000 | "Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch" on page 2052 |
| 4A0 | 4 | 000002A0 | "Port Status and Control USB2 (PORTSC3)—Offset 4A0h" on page 2053 |
| 4A4 | 4 | 00000000 | "Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h" on page 2055 |
| 4AC | 4 | 00000000 | "Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh" on page 2056 |
| 4B0 | 4 | 000002A0 | "Port Status and Control USB2 (PORTSC4)—Offset 4B0h" on page 2057 |
| 4B4 | 4 | 00000000 | "Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h" on page 2059 |
| 4BC | 4 | 00000000 | "Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh" on page 2060 |
| 4C0 | 4 | 000002A0 | "Port Status and Control USB3 (PORTSC5)—Offset 4C0h" on page 2061 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 4C4 | 4 | 00000000 | "Port Power Management Status and Control USB3 (PORTPMSC5)—Offset 4C4h" on page 2064 |
| 4C8 | 4 | 00000000 | "USB3 Port Link Info (PORTLI5)—Offset 4C8h" on page 2064 |
| 4D0 | 4 | 000002A0 | "Port Status and Control USB3 (PORTSC6)—Offset 4D0h" on page 2065 |
| 4D4 | 4 | 00000000 | "Port Power Management Status and Control USB3 (PORTPMSC6)—Offset 4D4h" on page 2068 |
| 4D8 | 4 | 00000000 | "USB3 Port Link Info (PORTLI6)—Offset 4D8h" on page 2068 |
| 4E0 | 4 | 000002A0 | "Port Status and Control USB3 (PORTSC7)—Offset 4E0h" on page 2069 |
| 4E4 | 4 | 00000000 | "Port Power Management Status and Control USB3 (PORTPMSC7)—Offset 4E4h" on page 2072 |
| 4E8 | 4 | 00000000 | "USB3 Port Link Info (PORTLI7)—Offset 4E8h" on page 2072 |
| 4F0 | 4 | 000002A0 | "Port Status and Control USB3 (PORTSC8)—Offset 4F0h" on page 2073 |
| 4F4 | 4 | 00000000 | "Port Power Management Status and Control USB3 (PORTPMSC8)—Offset 4F4h" on page 2076 |
| 4F8 | 4 | 00000000 | "USB3 Port Link Info (PORTLI8)—Offset 4F8h" on page 2076 |
| 2000 | 4 | 00000000 | "Microframe Index (RTMFINDEX)—Offset 2000h" on page 2076 |
| 2020 | 4 | 00000000 | "Interrupter 1 Management (IMAN0)—Offset 2020h" on page 2077 |
| 2024 | 4 | 00000FA0 | "Interrupter 1 Moderation (IMOD0)—Offset 2024h" on page 2077 |
| 2028 | 4 | 00000000 | "Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h" on page 2077 |
| 2030 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h" on page 2078 |
| 2034 | 4 | 00000000 | "Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h" on page 2078 |
| 2038 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h" on page 2079 |
| 203C | 4 | 00000000 | "Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch" on page 2079 |
| 2040 | 4 | 00000000 | "Interrupter 2 Management (IMAN1)—Offset 2040h" on page 2080 |
| 2044 | 4 | 00000FA0 | "Interrupter 2 Moderation (IMOD1)—Offset 2044h" on page 2080 |
| 2048 | 4 | 00000000 | "Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h" on page 2080 |
| 2050 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h" on page 2081 |
| 2054 | 4 | 00000000 | "Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h" on page 2081 |
| 2058 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h" on page 2081 |
| 205C | 4 | 00000000 | "Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch" on page 2082 |
| 2060 | 4 | 00000000 | "Interrupter 3 Management (IMAN2)—Offset 2060h" on page 2082 |
| 2064 | 4 | 00000FA0 | "Interrupter 3 Moderation (IMOD2)—Offset 2064h" on page 2082 |
| 2068 | 4 | 00000000 | "Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h" on page 2083 |
| 2070 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h" on page 2083 |
| 2074 | 4 | 00000000 | "Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h" on page 2083 |
| 2078 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h" on page 2084 |
| 207C | 4 | 00000000 | "Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch" on page 2084 |
| 2080 | 4 | 00000000 | "Interrupter 4 Management (IMAN3)—Offset 2080h" on page 2084 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2084 | 4 | 00000FA0 | "Interrupter 4 Moderation (IMOD3)—Offset 2084h" on page 2085 |
| 2088 | 4 | 00000000 | "Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h" on page 2085 |
| 2090 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h" on page 2085 |
| 2094 | 4 | 00000000 | "Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h" on page 2086 |
| 2098 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h" on page 2086 |
| 209C | 4 | 00000000 | "Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch" on page 2086 |
| 20A0 | 4 | 00000000 | "Interrupter 5 Management (IMAN4)—Offset 20A0h" on page 2087 |
| 20A4 | 4 | 00000FA0 | "Interrupter 5 Moderation (IMOD4)—Offset 20A4h" on page 2087 |
| 20A8 | 4 | 00000000 | "Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h" on page 2087 |
| 20B0 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h" on page 2088 |
| 20B4 | 4 | 00000000 | "Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h" on page 2088 |
| 20B8 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h" on page 2088 |
| 20BC | 4 | 00000000 | "Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh" on page 2089 |
| 20C0 | 4 | 00000000 | "Interrupter 6 Management (IMAN5)—Offset 20C0h" on page 2089 |
| 20C4 | 4 | 00000FA0 | "Interrupter 6 Moderation (IMOD5)—Offset 20C4h" on page 2089 |
| 20C8 | 4 | 00000000 | "Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h" on page 2090 |
| 20D0 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h" on page 2090 |
| 20D4 | 4 | 00000000 | "Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h" on page 2090 |
| 20D8 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h" on page 2091 |
| 20DC | 4 | 00000000 | "Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh" on page 2091 |
| 20E0 | 4 | 00000000 | "Interrupter 7 Management (IMAN6)—Offset 20E0h" on page 2091 |
| 20E4 | 4 | 00000FA0 | "Interrupter 7 Moderation (IMOD6)—Offset 20E4h" on page 2092 |
| 20E8 | 4 | 00000000 | "Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h" on page 2092 |
| 20F0 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h" on page 2092 |
| 20F4 | 4 | 00000000 | "Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h" on page 2093 |
| 20F8 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h" on page 2093 |
| 20FC | 4 | 00000000 | "Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh" on page 2093 |
| 2100 | 4 | 00000000 | "Interrupter 8 Management (IMAN7)—Offset 2100h" on page 2094 |
| 2104 | 4 | 00000FA0 | "Interrupter 8 Moderation (IMOD7)—Offset 2104h" on page 2094 |
| 2108 | 4 | 00000000 | "Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h" on page 2094 |
| 2110 | 4 | 00000000 | "Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h" on page 2095 |
| 2114 | 4 | 00000000 | "Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h" on page 2095 |
| 2118 | 4 | 00000000 | "Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h" on page 2095 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 211C | 4 | 00000000 | "Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch" on page 2096 |
| 3000 | 4 | 00000000 | "Door Bell 1 (DB0)—Offset 3000h" on page 2096 |
| 3004 | 4 | 00000000 | "Door Bell 2 (DB1)—Offset 3004h" on page 2097 |
| 3008 | 4 | 00000000 | "Door Bell 3 (DB2)—Offset 3008h" on page 2097 |
| 300C | 4 | 00000000 | "Door Bell 4 (DB3)—Offset 300Ch" on page 2098 |
| 3010 | 4 | 00000000 | "Door Bell 5 (DB4)—Offset 3010h" on page 2098 |
| 3014 | 4 | 00000000 | "Door Bell 6 (DB5)—Offset 3014h" on page 2099 |
| 3018 | 4 | 00000000 | "Door Bell 7 (DB6)—Offset 3018h" on page 2099 |
| 301C | 4 | 00000000 | "Door Bell 8 (DB7)—Offset 301Ch" on page 2100 |
| 3020 | 4 | 00000000 | "Door Bell 9 (DB8)—Offset 3020h" on page 2100 |
| 3024 | 4 | 00000000 | "Door Bell 10 (DB9)—Offset 3024h" on page 2101 |
| 3028 | 4 | 00000000 | "Door Bell 11 (DB10)—Offset 3028h" on page 2101 |
| 302C | 4 | 00000000 | "Door Bell 12 (DB11)—Offset 302Ch" on page 2102 |
| 3030 | 4 | 00000000 | "Door Bell 13 (DB12)—Offset 3030h" on page 2102 |
| 3034 | 4 | 00000000 | "Door Bell 14 (DB13)—Offset 3034h" on page 2103 |
| 3038 | 4 | 00000000 | "Door Bell 15 (DB14)—Offset 3038h" on page 2103 |
| 303C | 4 | 00000000 | "Door Bell 16 (DB15)—Offset 303Ch" on page 2104 |
| 3040 | 4 | 00000000 | "Door Bell 17 (DB16)—Offset 3040h" on page 2104 |
| 3044 | 4 | 00000000 | "Door Bell 18 (DB17)—Offset 3044h" on page 2105 |
| 3048 | 4 | 00000000 | "Door Bell 19 (DB18)—Offset 3048h" on page 2105 |
| 304C | 4 | 00000000 | "Door Bell 20 (DB19)—Offset 304Ch" on page 2106 |
| 3050 | 4 | 00000000 | "Door Bell 21 (DB20)—Offset 3050h" on page 2106 |
| 3054 | 4 | 00000000 | "Door Bell 22 (DB21)—Offset 3054h" on page 2107 |
| 3058 | 4 | 00000000 | "Door Bell 23 (DB22)—Offset 3058h" on page 2107 |
| 305C | 4 | 00000000 | "Door Bell 24 (DB23)—Offset 305Ch" on page 2108 |
| 3060 | 4 | 00000000 | "Door Bell 25 (DB24)—Offset 3060h" on page 2108 |
| 3064 | 4 | 00000000 | "Door Bell 26 (DB25)—Offset 3064h" on page 2109 |
| 3068 | 4 | 00000000 | "Door Bell 27 (DB26)—Offset 3068h" on page 2109 |
| 306C | 4 | 00000000 | "Door Bell 28 (DB27)—Offset 306Ch" on page 2110 |
| 3070 | 4 | 00000000 | "Door Bell 29 (DB28)—Offset 3070h" on page 2110 |
| 3074 | 4 | 00000000 | "Door Bell 30 (DB29)—Offset 3074h" on page 2111 |
| 3078 | 4 | 00000000 | "Door Bell 31 (DB30)—Offset 3078h" on page 2111 |
| 307C | 4 | 00000000 | "Door Bell 32 (DB31)—Offset 307Ch" on page 2112 |
| 3080 | 4 | 00000000 | "Door Bell 32 (DB32)—Offset 3080h" on page 2112 |
| 8004 | 4 | 20425355 | "XECP_SUPP_USB2_1—Offset 8004h" on page 2113 |
| 8008 | 4 | 30190401 | "XECP_SUPP_USB2_2—Offset 8008h" on page 2113 |
| 8024 | 4 | 20425355 | "XECP_SUPP_USB3_1—Offset 8024h" on page 2114 |
| 8028 | 4 | 10000405 | "XECP_SUPP_USB3_2—Offset 8028h" on page 2114 |
| 8070 | 4 | 004DFFC0 | "Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h" on page 2114 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 8078 | 4 | 00000000 | "Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h" on page 2115 |
| 807C | 4 | 00000000 | "Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch" on page 2115 |
| 8080 | 4 | 00000000 | "Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h" on page 2116 |
| 8094 | 4 | 00008100 | "Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h" on page 2117 |
| 80A4 | 4 | 002DFF90 | "Power Management Control (PMCTRL_REG)—Offset 80A4h" on page 2118 |
| 80A8 | 4 | 00315555 | "PGCB Control (PGCBCTRL_REG)—Offset 80A8h" on page 2120 |
| 80AC | 4 | 00000008 | "D0I3 Control (DOI3CTRL_REG)—Offset 80ACh" on page 2123 |
| 80B0 | 4 | 0001037F | "HOST_CTRL_MISC_REG—Offset 80B0h" on page 2124 |
| 80BC | 4 | 0000000F | "reg_SSPITPE_type (SSPITPE)—Offset 80BCh" on page 2126 |
| 80C0 | 4 | 015FC0F0 | "AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h" on page 2127 |
| 80C4 | 4 | 004A4008 | "Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h" on page 2130 |
| 80C8 | 4 | 0001A01F | "High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h" on page 2131 |
| 80CC | 4 | 00014080 | "Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh" on page 2131 |
| 80D0 | 4 | 00032010 | "System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h" on page 2132 |
| 80D4 | 4 | 00000000 | "Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h" on page 2133 |
| 80E0 | 4 | 808DBCA0 | "AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h" on page 2134 |
| 80E4 | 4 | 00000000 | "Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h" on page 2137 |
| 80EC | 4 | 18010000 | "SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh" on page 2138 |
| 80F0 | 4 | 310803A0 | "USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h" on page 2140 |
| 80F4 | 4 | 80C40620 | "USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h" on page 2142 |
| 80F8 | 4 | F865EB6B | "USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h" on page 2142 |
| 80FC | 4 | 00008003 | "USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh" on page 2143 |
| 810C | 4 | 00008020 | "Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch" on page 2143 |
| 8128 | 8 | 0F42528505647F42 | "USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h" on page 2144 |
| 8140 | 4 | 0A019132 | "Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h" on page 2144 |
| 8154 | 4 | 81390206 | "AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h" on page 2146 |
| 8164 | 4 | 000000FC | "USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h" on page 2148 |
| 8168 | 4 | 000282EE | "USB Power Gating Control (USB_PGC)—Offset 8168h" on page 2149 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 816C | 4 | 00000400 | "xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch" on page 2150 |
| 8170 | 4 | 96090032 | "USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h" on page 2152 |
| 8174 | 4 | 0040047D | "xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h" on page 2153 |
| 8178 | 4 | 000017FF | "xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h" on page 2154 |
| 817C | 4 | 00000000 | "xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch" on page 2154 |
| 8180 | 4 | 00000000 | "xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h" on page 2155 |
| 8184 | 4 | 00000000 | "xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h" on page 2155 |
| 8188 | 4 | 00F42F42 | "HOST_CTRL_BW_MAX3_REG—Offset 8188h" on page 2156 |
| 819C | 4 | 00000000 | "THRM_HOST_CTRL_REG—Offset 819Ch" on page 2157 |
| 81B4 | 4 | 0000000F | "THRM_HOST_CTRL_REG2—Offset 81B4h" on page 2158 |
| 81B8 | 4 | 000020C8 | "LFPSONCOUNT_REG—Offset 81B8h" on page 2158 |
| 81BC | 4 | 084204B0 | "reg_D0i2_CTRL_REG_type (D0i2_CTRL_REG)—Offset 81BCh" on page 2160 |
| 81C0 | 4 | 00000000 | "reg_D0i2_SCH_ALARM_CTRL_REG_type (D0i2_SCH_ALARM_CTRL_REG)—Offset 81C0h" on page 2162 |
| 81C4 | 4 | 00000000 | "reg_USB2PMCTRL_REG_type (USB2PMCTRL_REG)—Offset 81C4h" on page 2163 |
| 83F8 | 4 | 00000000 | "ECC_PARITY_ERROR_LOG_REG—Offset 83F8h" on page 2164 |
| 83FC | 4 | 00000000 | "ECC_POISONING_CTRL_REG—Offset 83FCh" on page 2166 |
| 8400 | 8 | 0000000000000000 | "USB2_PORT_STATE_REG—Offset 8400h" on page 2167 |
| 8408 | 8 | 0000000000000000 | "USB2_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h" on page 2167 |
| 8410 | 4 | 00010000 | "FUS1_REG—Offset 8410h" on page 2168 |
| 8414 | 4 | 00000000 | "FUS2_REG—Offset 8414h" on page 2169 |
| 841C | 4 | 00000000 | "STRAP1_REG—Offset 841Ch" on page 2169 |
| 8420 | 4 | 00000000 | "STRAP2_REG—Offset 8420h" on page 2169 |
| 8424 | 4 | 00000000 | "STRAP3_REG—Offset 8424h" on page 2170 |
| 8430 | 4 | 00000000 | "DFT_REG1—Offset 8430h" on page 2171 |
| 8434 | 4 | 00000000 | "DFT_REG2—Offset 8434h" on page 2172 |
| 8438 | 4 | 00000000 | "DFT_REG3—Offset 8438h" on page 2173 |
| 843C | 4 | 00000000 | "dft_reg4 (DFT_REG4)—Offset 843Ch" on page 2173 |
| 8440 | 4 | 00000000 | "dft_reg5 (DFT_REG5)—Offset 8440h" on page 2174 |
| 8448 | 4 | 00000000 | "XECP_CMDM_STS0—Offset 8448h" on page 2175 |
| 8450 | 4 | 00000000 | "XECP_CMDM_STS2—Offset 8450h" on page 2177 |
| 8454 | 4 | 00000000 | "XECP_CMDM_STS3—Offset 8454h" on page 2177 |
| 8460 | 4 | 00000000 | "AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h" on page 2177 |
| 8464 | 4 | 000D0000 | "Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h" on page 2178 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 846C | 4 | 00002201 | "USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch" on page 2179 |
| 84F8 | 4 | 00000000 | "USB2 Port Disable Override (USB2PDO)—Offset 84F8h" on page 2180 |
| 84FC | 4 | 00000000 | "USB3 Port Disable Override (USB3PDO)—Offset 84FCh" on page 2180 |
| 8604 | 2 | 0000 | "Command (CMD_MMIO)—Offset 8604h" on page 2181 |
| 8606 | 2 | 0290 | "Device Status (STS_MMIO)—Offset 8606h" on page 2182 |
| 8608 | 1 | 00 | "Revision ID (RID_MMIO)—Offset 8608h" on page 2184 |
| 8609 | 1 | 30 | "Programming Interface (PI_MMIO)—Offset 8609h" on page 2184 |
| 860A | 1 | 03 | "Sub Class Code (SCC_MMIO)—Offset 860Ah" on page 2184 |
| 860B | 2 | 0C | "Base Class Code (BCC_MMIO)—Offset 860Bh" on page 2185 |
| 860D | 1 | 00 | "Master Latency Timer (MLT_MMIO)—Offset 860Dh" on page 2185 |
| 860E | 2 | 00 | "Header Type (HT_MMIO)—Offset 860Eh" on page 2185 |
| 8610 | 8 | 0000000000000004 | "Memory Base Address (MBAR_MMIO)—Offset 8610h" on page 2186 |
| 862C | 2 | 0000 | "USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch" on page 2186 |
| 862E | 2 | 0000 | "USB Subsystem ID (SSID_MMIO)—Offset 862Eh" on page 2187 |
| 8634 | 1 | 70 | "Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h" on page 2187 |
| 863C | 1 | 00 | "Interrupt Line (ILINE_MMIO)—Offset 863Ch" on page 2187 |
| 863D | 3 | 00 | "Interrupt Pin (IPIN_MMIO)—Offset 863Dh" on page 2188 |
| 8640 | 4 | 000001FD | "XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h" on page 2189 |
| 8644 | 4 | 0003C000 | "XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h" on page 2191 |
| 8650 | 4 | 00000000 | "Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h" on page 2193 |
| 8658 | 4 | 00000000 | "Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h" on page 2196 |
| 8660 | 1 | 30 | "Serial Bus Release Number (SBRN_MMIO)—Offset 8660h" on page 2196 |
| 8661 | 1 | 60 | "Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h" on page 2197 |
| 8662 | 1 | 00 | "Best Effort Service Latency (BESL_MMIO)—Offset 8662h" on page 2197 |
| 8670 | 1 | 01 | "PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h" on page 2198 |
| 8671 | 1 | 80 | "Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h" on page 2198 |
| 8672 | 2 | C1C2 | "Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h" on page 2199 |
| 8674 | 2 | 0008 | "Power Management Control/Status (PM_CS_MMIO)—Offset 8674h" on page 2200 |
| 8680 | 1 | 05 | "Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h" on page 2201 |
| 8681 | 1 | 00 | "Next item pointer (MSI_NEXT_MMIO)—Offset 8681h" on page 2201 |
| 8682 | 2 | 0086 | "Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h" on page 2202 |
| 8684 | 4 | 00000000 | "Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h" on page 2202 |
| 8688 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h" on page 2203 |
| 868C | 2 | 0000 | "Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch" on page 2203 |

## Table 51-2. Summary of Memory Mapped I/O Registers—MBAR (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 8690 | 4 | F0140009 | "Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h" on page 2204 |
| 8694 | 4 | 01400010 | "Vendor Specific Header (VSHDR_MMIO)—Offset 8694h" on page 2204 |
| 8698 | 4 | 00000000 | "SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h" on page 2205 |
| 869C | 4 | 00080AC1 | "Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch" on page 2205 |
| 86A0 | 2 | 0800 | "Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h" on page 2206 |
| 86A4 | 4 | 00002000 | "High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h" on page 2207 |
| 86B0 | 4 | 00000000 | "XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h" on page 2208 |
| 86D0 | 4 | 00000000 | "XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h" on page 2208 |
| 86FC | 4 | 00000002 | "XHCC3 (XHCC3_MMIO)—Offset 86FCh" on page 2209 |
| 8700 | 4 | 0005000A | "Debug Capability ID Register (DCID)—Offset 8700h" on page 2209 |
| 8704 | 4 | 00000000 | "Debug Capability Doorbell Register (DCDB)—Offset 8704h" on page 2210 |
| 8708 | 4 | 00000000 | "Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h" on page 2210 |
| 8710 | 8 | 0000000000000000 | "Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h" on page 2211 |
| 8718 | 8 | 0000000000000000 | "Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h" on page 2211 |
| 8720 | 4 | 00000000 | "Debug Capability Control Register (DCCTRL)—Offset 8720h" on page 2212 |
| 8724 | 4 | 00000000 | "Debug Capability Status Register (DCST)—Offset 8724h" on page 2213 |
| 8728 | 4 | 00000080 | "Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h" on page 2214 |
| 8730 | 8 | 0000000000000000 | "Debug Capability Context Pointer Register (DCCP)—Offset 8730h" on page 2216 |
| 8738 | 4 | 80870000 | "Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h" on page 2216 |
| 873C | 4 | 00000000 | "Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch" on page 2217 |
| 8740 | 4 | 000030C3 | "Debug Capability Descriptor Parameters (DCDP)—Offset 8740h" on page 2217 |
| 8748 | 4 | 00000000 | "Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h" on page 2218 |
| 8760 | 4 | 00000000 | "DBC Control Register 1 (DBCCTL_REG)—Offset 8760h" on page 2220 |

## 51.2 Registers in Configuration Space

### 51.2.1 Vendor ID (VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** Reserved.<br><br>**Power Well:** Core |

### 51.2.2 Device ID (DID)—Offset 2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 2h

**Default:** 19D0h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x19d0<br>RO/V | **Device ID (DID):**<br><br>**Power Well:** Core |

## 51.2.3 Command (CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller<br>***Note:*** The corresponding Interrupt Status bit is not affected by the interrupt enable.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBE):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **SERR# Enable (SERR):** When set to 1, the XHC is capable of generating (internally) SERR#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Wait Cycle Control (WCC):** Reserved.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification.<br>***Note:*** This applies to both requests and completions from the system interface.<br>This bit must be set in order for the parity errors to generate SERR#.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VPS):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Memory Write Invalidate (MWI):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** When set, it allows XHC to act as a bus master.<br>When cleared, it disable XHC from initiating transactions on the system bus.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers.<br>If this bit is set, accesses to the XHC registers are enabled.<br>The Base Address register for the XHC should be programmed before this bit is set.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |

## 51.2.4 Device Status (STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 6h

**Default:** 0290h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **Detected Parity Error (DPE):** This bit is set by the SoC whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions.<br>Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW/1C | **Signaled System Error (SSE):** This bit is set by the SoC whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW/1C | **Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access.<br>This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW/1C | **Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access.<br>This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW/1C | **Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.<br><br>**Power Well:** Core |
| 10:9 | 0x1<br>RO | **DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion.<br>Read-Only.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW/1C | **Master Data Parity Error Detected (MDPED):** This bit is set by the SoC whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1.<br>Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 7 | 0x1<br>RO | **Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RO | **User Definable Features (UDF):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **66 MHz Capable (MC):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic.<br>This bit is a 1 when the interrupt is asserted.<br>This bit will be 0 when the interrupt is deasserted.<br>The value reported in this bit is independent of the value in the Interrupt Enable bit.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

### 51.2.5 Revision ID (RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):**<br><br>**Power Well:** Core |

### 51.2.6 Programming Interface (PI)—Offset 9h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 9h

**Default:** 30h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x30<br>RO | **Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.<br><br>**Power Well:** Core |

### 51.2.7 Sub Class Code (SCC)—Offset Ah

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + Ah

**Default:** 03h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x3<br>RO | **Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.<br><br>**Power Well:** Core |

### 51.2.8 Base Class Code (BCC)—Offset Bh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + Bh

**Default:** 0Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0xc<br>RO | **Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.<br><br>**Power Well:** Core |

## 51.2.9 Master Latency Timer (MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer.<br>The bits will be fixed at 0.<br><br>**Power Well:** Core |

## 51.2.10 Header Type (HT)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Multi-Function Bit (MFB):** Read only indicating single function device.<br><br>**Power Well:** Core |
| 6:0 | 0x0<br>RO | **Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.<br><br>**Power Well:** Core |

## 51.2.11 Memory Base Address (MBAR)—Offset 10h

Value in this register will be different after the enumeration process.

**Type:** PCI Configuration Register (Size: 64 bits)

**Offset:** [B:0, D:21, F:0] + 10h

**Default:** 0000000000000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:16 | 0x0<br>RW | **Base Address (BA):** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.<br><br>**Power Well:** Core |
| 15:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. Read-Only 0, this indicates that this function is requesting an 64KB block of memory.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Prefetchable:** This bit is hardwired to 0 indicating that this range should not be pre-fetched.<br><br>**Power Well:** Core |
| 2:1 | 0x2<br>RO | **Type:** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space.<br>If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space.<br><br>**Power Well:** Core |

## 51.2.12 USB Subsystem Vendor ID (SSVID)—Offset 2Ch

This register is modified and maintained by BIOS

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 2Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.<br><br>**Power Well:** Core |

## 51.2.13 USB Subsystem ID (SSID)—Offset 2Eh

This register is modified and maintained by BIOS

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 2Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).<br><br>**Power Well:** Core |

## 51.2.14 Capabilities Pointer (CAP_PTR)—Offset 34h

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 34h

**Default:** 70h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x70<br>RO | **Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.<br><br>**Power Well:** Core |

## 51.2.15 Interrupt Line (ILINE)—Offset 3Ch

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 3Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** This data is not used by the SoC. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.<br><br>**Power Well:** Core |

## 51.2.16 Interrupt Pin (IPIN)—Offset 3Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 3Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/L | **Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).<br><br>**Power Well:** Core |

## 51.2.17 XHC System Bus Configuration 1 (XHCC1)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 40h

**Default:** 000001FDh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/O | **Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits.<br>When set to '1' the write access to bits locked by this bit is disabled (locked state).<br>When set to '0', the write access to bit locked by this bit is enabled (unlocked state).<br>Writable once after platform reset.<br><br>**Power Well:** Core |
| 30:25 | 0x0<br>RW | **ECO1:**<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.<br><br>**Power Well:** Core |
| 23 | 0x0<br>RW/C | **Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle.<br>Cleared by SW when the bit is written with value of '1'.<br><br>**Power Well:** Core |
| 22 | 0x0<br>RW | **Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.<br><br>**Power Well:** Core |
| 21:19 | 0x0<br>RW | **Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified.<br>• 000: Disabled<br>• 001: 32 bb_cclk<br>• 010: 64 bb_cclk<br>• 011: 128 bb_cclk<br>• 100: 256 bb_cclk<br>• 101: 512 bb_cclk<br>• 110: 1024 bb_cclk<br>• 111: 131072 bb_cclk<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RW | **XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power management to be enabled.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW | **D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power management to be enables.<br>This bit can only be set if the XHCI L1 Override P2 chicken bit is set.<br><br>**Power Well:** Core |
| 16:12 | 0x0<br>RW | **Periodic Complete Pre Wake Time (PCPWT):** This allows for platform wake time before the next scheduled periodic transaction.<br>The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert.<br>This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame.<br>Register Format: Bits (16:12) represents the # of bytes remaining with a 256B granularity.<br>Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated.<br>This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected.<br>This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.<br><br>**Power Well:** Core |
| 10:8 | 0x1<br>RW | **L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC.<br>• 000: Disabled<br>• 001: 128 bb_cclk<br>• 010: 256 bb_cclk<br>• 011: 512 bb_cclk<br>• 100: 1024 bb_cclk<br>• 101: 2048 bb_cclk<br>• 110: 4096 bb_cclk<br>• 111: 131072 bb_cclk<br><br>**Power Well:** Core |
| 7:6 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type.<br><br>**Power Well:** Core |
| 5:4 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type.<br><br>**Power Well:** Core |
| 3:2 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type.<br><br>**Power Well:** Core |
| 1:0 | 0x1<br>RW | **Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter.<br><br>**Power Well:** Core |

## 51.2.18 XHC System Bus Configuration 2 (XHCC2)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 44h

**Default:** 0003C000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration.<br>It must be set by BIOS after configuration of the OC mapping bits is complete.<br>Once this bit is set, OC mapping shall not be changed by SW.<br><br>**Power Well:** SUS |
| 30:26 | 0x0<br>RW | **ECO1:**<br>**Power Well:** Core |
| 25 | 0x0<br>RW | **DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request.<br>• 0: 4KB<br>• 1: 64B<br><br>**Power Well:** Core |
| 24:22 | 0x0<br>RW | **IDMA Write Request Size Control (IDMA_WRREQSZCTRL):** Write Request Size Control:<br>This bit controls the maximum size of each Write Request.<br>• 000: 128B<br>• 001: 256B<br>• 011 - 110: Reserved<br>• 111: 64B<br><br>**Power Well:** Core |
| 21 | 0x0<br>RW | **XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads.<br>• 0 - xHC will clear RO for all upstream read requests.<br>• 1 - xHC will set RO for all upstream read requests.<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW | **IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.<br><br>**Power Well:** Core |
| 19:14 | 0xf<br>RW | **Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource.<br>This value is zero based.<br>000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation.<br>If set less than default allows under-allocation.<br>Only allowed to be programmed when BME = 0 and no outstanding downstream completion.<br><br>**Power Well:** Core |
| 13:12 | 0x0<br>RW | **SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit.<br>• 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state<br>• 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW.<br>• 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI.<br>• 11b: Reserved<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x0<br>RW | **MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write.<br>• 0b (Default): Delay MMIO Read after MMIO Write<br>• 1b: Do not delay MMIO Read after MMIO Write Note that this delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write.<br>• 0b (Default): Do not delay MMIO Write after previous MMIO Write<br>• 1b: Delay MMIO Write after previous MMIO Write<br>Note that the delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.<br><br>**Power Well:** Core |
| 9:8 | 0x0<br>RW | **SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave.<br>• 00: Never inhibit Cx<br>• 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior)<br>• 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1<br>• 11: Always inhibit Cx<br><br>**Power Well:** Core |
| 7:6 | 0x0<br>RW | **SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave.<br>• 00: Never inhibit DMI L1.<br>• 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior).<br>• 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1.<br>• 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.<br><br>**Power Well:** Core |
| 5:3 | 0x0<br>RW | **L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified.<br>If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake,<br>• 000: Disabled<br>• 001: 128 bb_cclk<br>• 010: 256 bb_cclk<br>• 011: 512 bb_cclk<br>• 100: 1024 bb_cclk<br>• 101: 2048 bb_cclk<br>• 110: 4096 bb_cclk<br>• 111: 131072 bb_cclk<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Read Request Size Control (RDREQSZCTRL):** Read Request Size Control:<br>This bit controls the maximum size of each Read Request.<br>• 000: 128B<br>• 001: 256B<br>• 010: 512B<br>• 011 - 110: Reserved<br>• 111: 64B<br><br>**Power Well:** Core |

## 51.2.19 Clock Gating (XHCLKGTEN)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 50h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Rsvd2:** Reserved<br><br>**Power Well:** Core |
| 28 | 0x0<br>RW | **Naking USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not.<br>• 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation.<br>• 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met<br>• 0 - Disallow SRAM Power Gating.<br>• 1 - Allow SRAM Power Gating<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW | **SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports on top of trunk clock gating.<br>• 0 - P3 state NOT allowed to result in PXP PLL shutdown.<br>• 1- P3 state allowed to result in PXP PLL shutdown<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request.<br>***Note:*** If USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.<br><br>**Power Well:** Core |
| 23:20 | 0x0<br>RW | **HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted.<br>• (0) ==) U0 or deeper<br>• (1) ==) NA (no support for U1)<br>• (2) ==) U2 (L1) or deeper<br>• (3) ==) U3 (L2) or deeper<br><br>**Power Well:** Core |
| 19:16 | 0x0<br>RW | **SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating.<br>• 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state.<br>• 1 - xHC may allow frame timer to be gated regardless of EU3S.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW | **XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated.<br>• 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running.<br>• 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB.<br>Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock.<br>• 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle.<br>• 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW | **XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is at least 1 non Remote Wake Enabled HS Port in U2.<br>• 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2.<br>• 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.<br><br>**Power Well:** Core |
| 11:10 | 0x0<br>RW | **XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports.<br>Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted.<br>• (0) ==) L1 or deeper<br>• (1) ==) L2 or deeper<br><br>**Power Well:** Core |
| 9:8 | 0x0<br>RW | **HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions.<br>PLL Shutdown is allowed in:<br>• 00b Disabled (Link states shall be disabled for DMI PLL shutdown)<br>• 01b U0 or conditions for 10b setting.<br>• 10b U2 or conditions for 11b setting.<br>• 10b U3, Disconnected, Disabled or Powered-Off.<br><br>**Power Well:** Core |
| 7:5 | 0x0<br>RW | **SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions.<br>PLL Shutdown is allowed in:<br>• 000b Disabled (Link states shall be ignored for DMI PLL shutdown).<br>• 001b U0 or conditions for 010b setting<br>• 010b U1 or conditions for 011b setting<br>• 011b U2 or conditions for 100b setting<br>• 100b U3, Disconnected, Disabled or Powered-Off<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW | **XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met.<br>***Note:*** If XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met.<br>***Note:*** If XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met.<br>***Note:*** If XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met.<br>***Note:*** If XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |

## 51.2.20 Audio Time Synchronization (AUDSYNC)—Offset 58h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 58h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 29:16 | 0x0<br>RO/V | **Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal.<br>Bits (29:16) reflect state of bits (13:0) of FRINDEX<br><br>**Power Well:** Core |
| 15:13 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RO/V | **Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame.<br>Captured value represents number of 8 high-speed bit time units from start of micro-frame.<br>At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end.<br>Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.<br><br>**Power Well:** Core |

## 51.2.21 Serial Bus Release Number (SBRN)—Offset 60h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 60h

**Default:** 30h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x30<br>RO | **Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.<br><br>**Power Well:** SUS |

## 51.2.22 Frame Length Adjustment (FLADJ)—Offset 61h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 61h

**Default:** 60h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):**<br><br>**Power Well:** SUS |
| 6 | 0x1<br>RO | **No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.<br><br>**Power Well:** Core |
| 5:0 | 0x20<br>RO | **Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field.<br>The default value is decimal 32 (20h), which gives a SOF cycle time of 60000.<br>Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal)<br>• 59488 0 (00h)<br>• 59504 1 (01h)<br>• 59520 2 (02h)<br>• ...<br>• 59984 31 (1Fh)<br>• 60000 32 (20h)<br>• ...<br>• 60480 62 (3Eh)<br>• 60496 63 (3Fh)<br>Each decimal value change to this register corresponds to 16 high-speed bit times.<br>The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field.<br>The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value<br><br>**Power Well:** SUS |

## 51.2.23 Best Effort Service Latency (BESL)—Offset 62h

Best Effort Service Latency.

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 62h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW/L | **Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field.<br>This is programmed by BIOS based on platform parameters.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW/L | **Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field.<br>This is programmed by BIOS based on platform parameters.<br><br>**Power Well:** Core |

## 51.2.24 PCI Power Management Capability ID (PM_CID)—Offset 70h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 70h

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RO | **PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.<br><br>**Power Well:** Core |

## 51.2.25 Next Item Pointer #1 (PM_NEXT)—Offset 71h

This register is modified and maintained by BIOS

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 71h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x80<br>RW/L | **Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h.<br>This register is writable when the Access Control bit is set to '0'.<br>This allows BIOS to effectively hide the next capability registers, if necessary.<br>This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic.<br>Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden.<br>***Note:*** This value is never expected to be programmed.<br><br>**Power Well:** Core |

## 51.2.26 Power Management Capabilities (PM_CAP)—Offset 72h

Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the SoC is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 72h

**Default:** C1C2h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x18<br>RW/L | **PME_Support:** This 5-bit field indicates the power states in which the function may assert PME#.<br>The SoC XHC does not support the D1 or D2 states.<br>For all other states, the SoC XHC is capable of generating PME#.<br>Software should never need to modify this field.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW/L | **D2_Support:** The D2 state is not supported.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RW/L | **D1_Support:** The D1 state is not supported.<br><br>**Power Well:** Core |
| 8:6 | 0x7<br>RW/L | **Aux_Current:** The SoC XHC reports 375mA maximum Suspend well current required when in the D3cold state.<br>This value can be written by BIOS when a more accurate value is known.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW/L | **DSI:** The SoC reports 0, indicating that no device-specific initialization is required.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/L | **PME Clock (PMEClock):** The SoC reports 0, indicating that no PCI clock is required to generate PME#.<br><br>**Power Well:** Core |
| 2:0 | 0x2<br>RW/L | **Version:** The SoC reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.<br><br>**Power Well:** Core |

## 51.2.27 Power Management Control/Status (PM_CS)—Offset 74h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 74h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **PME_Status:** This bit is set when the SoC XHC would normally assert the PME# signal independent of the state of the PME_En bit.<br>Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled).<br>Writing a 0 has no effect.<br>This bit must be explicitly cleared by the operating system each time the operating system is loaded.<br><br>**Power Well:** SUS |
| 14:13 | 0x0<br>RO | **RSVD:** Reserved<br><br>**Power Well:** Core |
| 12:9 | 0x0<br>RO | **RSVD:** Reserved<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **PME_En:** A 1 enables the SoC XHC to generate an internal PME signal when PME_Status is 1.<br>This bit must be explicitly cleared by the operating system each time it is initially loaded.<br><br>**Power Well:** SUS |
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **Reserved (RSVD2):** Reserved.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RW | **PowerState:** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state.<br>The definition of the field values are:<br>• 00b - D0 state<br>• 11b - D3hot state<br><br>**Power Well:** Core |

## 51.2.28 Message Signaled Interrupt CID (MSI_CID)—Offset 80h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 80h

**Default:** 05h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates that this is an MSI capability.<br><br>**Power Well:** Core |

## 51.2.29 Next item pointer (MSI_NEXT)—Offset 81h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:21, F:0] + 81h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates that this is the last item on the capability list.<br><br>**Power Well:** Core |

## 51.2.30 Message Signaled Interrupt Message Control (MSI_MCTL)—Offset 82h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 82h

**Default:** 0086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RO | **Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported<br><br>**Power Well:** Core |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RW | **Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.<br><br>**Power Well:** Core |
| 3:1 | 0x3<br>RO | **Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert.<br>This field must be set by HW to reflect the number of Interrupters supported.<br>Encoding number of Vectors requested (number of Interrupters)<br>• 000 1<br>• 001 2<br>• 010 4<br>• 011 8<br>• 100 16<br>• 101 32<br>• 110-111 Reserved<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts.<br>If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.<br><br>**Power Well:** Core |

## 51.2.31 Message Signaled Interrupt Message Address (MSI_MAD)—Offset 84h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Addr:** Lower DW of system specified message address, always DWORD aligned.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 51.2.32 Message Signaled Interrupt Upper Address (MSI_MUAD)—Offset 88h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Addr (UpperAddr):** Upper DW of system specified message address.<br><br>**Power Well:** Core |

## 51.2.33 Message Signaled Interrupt Message Data (MSI_MD)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + 8Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data:** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction.<br><br>**Power Well:** Core |

## 51.2.34 Device Idle Capability (DEVIDLE)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 90h

**Default:** F0140009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf<br>RO | **VID:** Reserved.<br><br>**Power Well:** Core |
| 27:24 | 0x0<br>RO | **REV:** Reserved.<br><br>**Power Well:** Core |
| 23:16 | 0x14<br>RO | **Length (LENGTH):** Indicates that this capability is 16 bytes long.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.<br><br>**Power Well:** Core |
| 7:0 | 0x9<br>RO | **Capability ID (CID):** Reserved.<br><br>**Power Well:** Core |

## 51.2.35 Vendor Specific Header (VSHDR)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 94h

**Default:** 01400010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x14<br>RO | **VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register<br><br>**Power Well:** Core |
| 19:16 | 0x0<br>RO | **VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.<br><br>**Power Well:** Core |
| 15:0 | 0x10<br>RO | **VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.<br><br>**Power Well:** Core |

## 51.2.36 SW LTR POINTER (SWLTRPTR)—Offset 98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR.<br>The value of this register is a don't care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 3:1 | 0x0<br>RO | **Base Address Register Number (BARNUM):** Contains the 0s based AR Number of the BAR which contains the location of the SW LTR MMIO register.<br>When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number.<br>Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Valid (VALID):** Set to 1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM.<br>Set to 0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition.<br>This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.<br><br>**Power Well:** Core |

## 51.2.37 Device Idle Pointer Register (DEVIDLEPTR)—Offset 9Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + 9Ch

**Default:** 00080AC1h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x80ac<br>RW/L | **DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR.<br>The value of this register is a don't care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 3:1 | 0x0<br>RO | **Base Address Register Number (BARNUM):** Contains the 0s based BAR Number of the BAR which contains the location of the DevIdle MMIO register.<br>When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number.<br>Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RW/L | **Valid (VALID):** Set to 1 to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM.<br>Set to 0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition.<br>This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.<br><br>**Power Well:** Core |

## 51.2.38 Device Idle Power ON Latency (DEVIDLEPOL)—Offset A0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:21, F:0] + A0h

**Default:** 0800h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 12:10 | 0x2<br>RW/L | **Power On Latency Scale (POLS):** Latency Scale multiplier:<br>• 010: 1us<br>• 011: 32us<br>• All other settings are reserved.<br>The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms.<br>This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value.<br>This register is undefined if the DevIdle.Valid bit is 0.<br><br>**Power Well:** Core |
| 9:0 | 0x0<br>RW/L | **Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us.<br>This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value.<br>This register is undefined if the DevIdle.Valid bit is 0.<br><br>**Power Well:** Core |

## 51.2.39 High Speed Configuration 2 (HSCFG2)—Offset A4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + A4h

**Default:** 00002000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RW | **PORT1_HOST_MODE_OVERRIDE:** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.<br><br>**Power Well:** Core |
| 17:16 | 0x0<br>RW | **eUSB2SEL:** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2<br>• 0: Port is mapped to USB2<br>• 1: Port is mapped to eUSB2<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ignore IN EP s.<br>• 0 HS ASYNC Active will include IN EP s.<br>• 1 HS ASYNC Active will mask/ignore IN EP s.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW | **HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK.<br>• 0 HS OUT ASYNC Active will include EP s that are polling.<br>• 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.<br><br>**Power Well:** Core |
| 13 | 0x1<br>RW | **HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK.<br>• 0 HS IN ASYNC Active will include EP s that are polling.<br>• 1 HS IN ASYNC Active will mask/ignore EP s that are polling.<br><br>**Power Well:** Core |
| 12:11 | 0x0<br>RW | **HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active.<br>This will determine how the smallest service interval among active EP s and number of active EP s are used.<br>• 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement.<br>• 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement.<br>• 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication.<br>• 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indication<br><br>**Power Well:** Core |
| 10:4 | 0x0<br>RW | **HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Active may include HS/FS/LS INTR IN EP active indication.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Active will include HS/FS/LS INTR IN EP active indication.<br><br>**Power Well:** Core |

## 51.2.40 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1)—Offset B0h

The RW/L property of this register is controlled by OCCFDONE bit.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Rsvd:** Reserved<br><br>**Power Well:** SUS |
| 3:0 | 0x0<br>RW/L | **OC Mapping (OCM):** Reserved.<br><br>**Power Well:** SUS |

## 51.2.41 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1)—Offset D0h

The RW/L property of this register is controlled by OCCFDONE bit.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW/L | **OC Mapping (OCM):** Reserved.<br><br>**Power Well:** SUS |

## 51.2.42 XHCC3—Offset FCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:21, F:0] + FCh

**Default:** 00000002h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (Rsvd1):** Reserved<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface<br><br>**Power Well:** SUS |
| 1 | 0x1<br>RW | **Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW/L | **Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.<br><br>**Power Well:** SUS |

## 51.3 Registers in Memory Space—MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) MBAR described in Section 51.2.11, "Memory Base Address (MBAR)—Offset 10h" on page 2012.

### 51.3.1 Capability Registers Length (CAPLENGTH)—Offset 0h

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 80h

**BAR and Offset:** [MBAR] + 0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x80<br>RW/L | **Capability Registers Length (CAPLENGTH):** Reserved.<br><br>**Power Well:** Core |

### 51.3.2 Host Controller Interface Version Number (HCIVERSION)—Offset 2h

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0100h

**BAR and Offset:** [MBAR] + 2h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x100<br>RW/L | **Host Controller Interface Version Number (HCIVERSION):** Reserved.<br><br>**Power Well:** Core |

## 51.3.3 Structural Parameters 1 (HCSPARAMS1)—Offset 4h

This register is modified and maintained by BIOS.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 08000820h

**BAR and Offset:** [MBAR] + 4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x8 RW/L | **Number of Ports (MaxPorts):** Reserved.<br>**Power Well:** Core |
| 23:19 | 0x0 RW/L | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 18:8 | 0x8 RW/L | **Number of Interrupters (MaxIntrs):** Reserved.<br>**Power Well:** Core |
| 7:0 | 0x20 RW/L | **Number of Device Slots (MaxSlots):** Reserved.<br>**Power Well:** Core |

## 51.3.4 Structural Parameters 2 (HCSPARAMS2)—Offset 8h

This register is modified and maintained by BIOS.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 94000054h

**BAR and Offset:** [MBAR] + 8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x12 RW/L | **Max Scratchpad Buffers LO (MaxScratchpadBufs):** Reserved.<br>**Power Well:** Core |
| 26 | 0x1 RW/L | **Scratchpad Restore (SPR):** Reserved.<br>**Power Well:** Core |
| 25:21 | 0x0 RW/L | **Max Scratchpad Buffers HI (MaxScratchpadBufs_HI):** Reserved.<br>**Power Well:** Core |
| 20:8 | 0x0 RW/L | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 7:4 | 0x5 RW/L | **Event Ring Segment Table Max (ERSTMax):** Reserved.<br>**Power Well:** Core |
| 3:0 | 0x4 RW/L | **Isochronous Scheduling Threshold (IST):** Reserved.<br>**Power Well:** Core |

## 51.3.5 Structural Parameters 3 (HCSPARAMS3)—Offset Ch

This register is modified and maintained by BIOS.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00040001h

**BAR and Offset:** [MBAR] + Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x4<br>RW/L | **U2 Device Exit Latency (U2DEL):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RW/L | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x1<br>RW/L | **U1 Device Exit Latency (U1DEL):** Reserved.<br><br>**Power Well:** Core |

## 51.3.6 Capability Parameters (HCCPARAMS)—Offset 10h

This register is modified and maintained by BIOS.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 10h

**Default:** 200077C1h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x2000<br>RW/L | **xHCI Extended Capabilities Pointer (xECP):** Reserved.<br>**Power Well:** Core |
| 15:12 | 0x7<br>RW/L | **Maximum Primary Stream Array Size (MaxPSASize):** Reserved.<br>**Power Well:** Core |
| 11 | 0x0<br>RW/L | **Contiguous Frame ID Capability (CFC):** Reserved.<br>**Power Well:** Core |
| 10 | 0x1<br>RW/L | **Stopped EDLTA Capabilty (SEC):** Reserved.<br>**Power Well:** Core |
| 9 | 0x1<br>RW/L | **Stopped - Short Packet Capability (SPC):** Reserved.<br>**Power Well:** Core |
| 8 | 0x1<br>RW/L | **Parst All Event Data (PAE):** Reserved.<br>**Power Well:** Core |
| 7 | 0x1<br>RW/L | **No Secondary SID Support (NSS):** Reserved.<br>**Power Well:** Core |
| 6 | 0x1<br>RW/L | **Latency Tolerance Messaging Capability (LTC):** Reserved.<br>**Power Well:** Core |
| 5 | 0x0<br>RW/L | **Light HC Reset Capability (LHRC):** Reserved.<br>**Power Well:** Core |
| 4 | 0x0<br>RW/L | **Port Indicators (PIND):** Reserved.<br>**Power Well:** Core |
| 3 | 0x0<br>RW/L | **Port Power Control (PPC):** Reserved.<br>**Power Well:** Core |
| 2 | 0x0<br>RW/L | **Context Size (CSZ):** Reserved.<br>**Power Well:** Core |
| 1 | 0x0<br>RW/L | **BW Negotiation Capability (BNC):** Reserved.<br>**Power Well:** Core |
| 0 | 0x1<br>RW/L | **64-bit Addressing Capability (AC64):** Reserved.<br>**Power Well:** Core |

## 51.3.7 Doorbell Offset (DBOFF)—Offset 14h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00003000h

**BAR and Offset:** [MBAR] + 14h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0xc00<br>RO | **Doorbell Array Offset (DBAO):** Reserved.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.8 Runtime Register Space Offset (RTSOFF)—Offset 18h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00002000h

**BAR and Offset:** [MBAR] + 18h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x100<br>RO | **Runtime Register Space Offset (RTRSO):** Reserved.<br><br>**Power Well:** Core |
| 4:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.9 USB Command (USBCMD)—Offset 80h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 80h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Enable U3 MFINDEX Stop (EU3S):** Reserved.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **Enable Wrap Event (EWE):** Reserved.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RW | **Controller Restore State (CRS):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Controller Save State (CSS):** Reserved.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **Light Host Controller Reset (LHCRST):** Reserved.<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Host System Error Enable (HSEE):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **Interrupter Enable (INTE):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Host Controller Reset (HCRST):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Run/Stop (RS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.10 USB Status (USBSTS)—Offset 84h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000001h

**BAR and Offset:** [MBAR] + 84h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:13 | 0x0<br>RO | **Rsvd3:** Reserved.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO | **Host Controller Error (HCE):** This bit is not preset in HC, this is deviation from XHCI 1.0 spec.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Controller Not Ready (CNR):** This is deviation from XHCI 1.0 spec.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW/C | **Save/Restore Error (SRE):** Reserved.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Restore State Status (RSS):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RO | **Save State Status (SSS):** Reserved.<br><br>**Power Well:** Core |
| 7:5 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/C | **Port Change Detect (PCD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Interrupt (EINT):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/C | **Host System Error (HSE):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RO | **HCHalted (HCH):** Reserved.<br><br>**Power Well:** Core |

## 51.3.11 Page Size (PAGESIZE)—Offset 88h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000001h

**BAR and Offset:** [MBAR] + 88h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x1<br>RO | **Page Size (PAGESIZE):** Reserved.<br><br>**Power Well:** Core |

## 51.3.12 Device Notification Control (DNCTRL)—Offset 94h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 94h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Notification Enable (N0_N15):** Reserved.<br><br>**Power Well:** Core |

## 51.3.13 Command Ring Low (CRCR_LO)—Offset 98h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 98h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Command Ring Pointer (CRP):** Reserved.<br><br>**Power Well:** Core |
| 5:4 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Command Ring Running (CRR):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/1S | **Command Abort (CA):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/1S | **Command Stop (CS):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Ring Cycle State (RCS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.14 Command Ring High (CRCR_HI)—Offset 9Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 9Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Command Ring Pointer (CRP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.15 Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + B0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Device Context Base Address Array Pointer (DCBAAP):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.16 Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + B4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Device Context Base Address Array Pointer (DCBAAP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.17 Configure (CONFIG)—Offset B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + B8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Max Device Slots Enabled (MaxSlotsEn):** Reserved.<br><br>**Power Well:** Core |

## 51.3.18 Port Status and Control USB2 (PORTSC1)—Offset 480h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 480h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** Reserved.<br><br>**Power Well:** Core |
| 29:28 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** Reserved.<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 19 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):** Reserved.<br><br>**Power Well:** SUS |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 9 | 0x1<br>RW | **Port Power (PP):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 8:5 | 0x5<br>RW | **Port Link State (PLS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):**<br>***Note:*** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.19 Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 484h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Port Test Control (PTC):** When this field is '0', the port is not operating in a test mode. (Default) A non-zero value indicates that the port is operating in test mode in indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Disabled state. If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error.<br>The encoding of the Test Mode bits for a USB 2.0 port are:<br>Value Test Mode<br>• 0h - Test mode not enabled<br>• 1h - Test J_STATE<br>• 2h - Test K_STATE<br>• 3h - Test SE0_NAK<br>• 4h - Test Packet<br>• 5h - Test FORCE_ENABLE<br>• 6h-14h - Reserved.<br>• 15 - Port Test Control Error |
| 27:17 | 0x0<br>RO | Reserved. |
| 16 | 0x0<br>RW | **Hardware LPM Enable (HLE):**<br>• 0 - Disable<br>• 1 - Enable. When this bit is a 1, hardware controlled LPM shall be enabled for this port. Refer to Section 4 of the USB 2.0 LPM Specification for more information. |
| 15:8 | 0x0<br>RO | Reserved. |
| 7:4 | 0x0<br>RW | **Best Effort Service Latency:** System software sets this field to indicate the recipient device how long the xHCI will driver resume if it initiates an exit from L1. |
| 3 | 0x0<br>RW | **Remote Wake Enable (RWE):** The host system sets this flag to enable or disable the device for remote wake from L1.<br>• 0 - disable<br>• 1 - enable<br>The value of this flag will temporarily (while in L1) override the current setting of the Remote Wake feature set by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, Revision 2.0, Chapter 9. |
| 2:0 | 0x0<br>RW | **L1 Status (L1S):** This field is used by software to determine whether an L1-based suspend request was successful. |

## 51.3.20 Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters necessary for xHC to automatically generate an LPM Token to the downstream device.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 48Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 13:10 | 0x0<br>RW | **Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1.<br>The HIRDD value is is encoded as follows:<br>• 0h: 50 us (default)<br>• 1h: 125 us<br>• 2h: 200 us<br>• ...<br>• Fh: 1.175ms<br>The value of 0h is interpreted as 50 us.<br>Each incrementing value adds 75 us to the previous value.<br><br>**Power Well:** SUS |
| 9:2 | 0x0<br>RW | **L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer.<br>This field shall be set to 00h by assertion of PR to '1'.<br>Following are permissible values:<br>• 00h: 128 us (default)<br>• 01h: 256 us<br>• ...<br>• FFh: 65,280us<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RW | **Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used.<br>Following are permissible values:<br>• 0: Initiate L1 using HIRD only time out (default)<br>• 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD<br>• 2: Reserved<br>• 3: Reserved<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.21 Port Status and Control USB2 (PORTSC2)—Offset 490h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 490h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** Reserved.<br><br>**Power Well:** Core |
| 29:28 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** Reserved.<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 19 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):** Reserved.<br><br>**Power Well:** SUS |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 9 | 0x1<br>RW | **Port Power (PP):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 8:5 | 0x5<br>RW | **Port Link State (PLS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.22 Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 494h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Port Test Control (PTC):** When this field is '0', the port is not operating in a test mode. (Default) A non-zero value indicates that the port is operating in test mode in indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Disabled state. If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error.<br>The encoding of the Test Mode bits for a USB 2.0 port are:<br>Value Test Mode<br>• 0h - Test mode not enabled<br>• 1h - Test J_STATE<br>• 2h - Test K_STATE<br>• 3h - Test SE0_NAK<br>• 4h - Test Packet<br>• 5h - Test FORCE_ENABLE<br>• 6h-14h - Reserved.<br>• 15 - Port Test Control Error |
| 27:17 | 0x0<br>RO | Reserved. |
| 16 | 0x0<br>RW | **Hardware LPM Enable (HLE):**<br>• 0 - Disable<br>• 1 - Enable. When this bit is a 1, hardware controlled LPM shall be enabled for this port. Refer to Section 4 of the USB 2.0 LPM Specification for more information. |
| 15:8 | 0x0<br>RO | Reserved. |
| 7:4 | 0x0<br>RW | **Best Effort Service Latency:** System software sets this field to indicate the recipient device how long the xHCI will driver resume if it initiates an exit from L1. |
| 3 | 0x0<br>RW | **Remote Wake Enable (RWE):** The host system sets this flag to enable or disable the device for remote wake from L1.<br>• 0 - disable<br>• 1 - enable<br>The value of this flag will temporarily (while in L1) override the current setting of the Remote Wake feature set by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, Revision 2.0, Chapter 9. |
| 2:0 | 0x0<br>RW | **L1 Status (L1S):** This field is used by software to determine whether an L1-based suspend request was successful. |

## 51.3.23 Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters necessary for xHC to automatically generate an LPM Token to the downstream device.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 49Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 13:10 | 0x0<br>RW | **Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1.<br>• The HIRDD value is encoded as follows:<br>• 0h: 50 us (default)<br>• 1h: 125 us<br>• 2h: 200 us<br>• ...<br>• Fh: 1.175ms<br>The value of 0h is interpreted as 50 us.<br>Each incrementing value adds 75 us to the previous value.<br><br>**Power Well:** SUS |
| 9:2 | 0x0<br>RW | **L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer.<br>This field shall be set to 00h by assertion of PR to '1'.<br>Following are permissible values:<br>• 00h: 128 us (default)<br>• 01h: 256 us.<br>• ...<br>• FFh: 65,280us<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RW | **Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used.<br>Following are permissible values:<br>• 0: Initiate L1 using HIRD only time out (default)<br>• 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD<br>• 2: Reserved<br>• 3: Reserved<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.24 Port Status and Control USB2 (PORTSC3)—Offset 4A0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 4A0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** Reserved.<br><br>**Power Well:** Core |
| 29:28 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** Reserved.<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 19 | 0x0<br>RW/C | **Reserved:** Reserved.<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):** Reserved.<br><br>**Power Well:** SUS |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 9 | 0x1<br>RW | **Port Power (PP):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 8:5 | 0x5<br>RW | **Port Link State (PLS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):**<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.25 Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4A4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Port Test Control (PTC):** When this field is '0', the port is not operating in a test mode. (Default) A non-zero value indicates that the port is operating in test mode in indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Disabled state. If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error.<br>The encoding of the Test Mode bits for a USB 2.0 port are:<br>Value Test Mode<br>• 0h - Test mode not enabled<br>• 1h - Test J_STATE<br>• 2h - Test K_STATE<br>• 3h - Test SE0_NAK<br>• 4h - Test Packet<br>• 5h - Test FORCE_ENABLE<br>• 6h-14h - Reserved.<br>• 15 - Port Test Control Error |
| 27:17 | 0x0<br>RO | Reserved. |
| 16 | 0x0<br>RW | **Hardware LPM Enable (HLE):**<br>• 0 - Disable<br>• 1 - Enable. When this bit is a 1, hardware controlled LPM shall be enabled for this port. Refer to Section 4 of the USB 2.0 LPM Specification for more information. |
| 15:8 | 0x0<br>RO | Reserved. |
| 7:4 | 0x0<br>RW | **Best Effort Service Latency:** System software sets this field to indicate the recipient device how long the xHCI will driver resume if it initiates an exit from L1. |
| 3 | 0x0<br>RW | **Remote Wake Enable (RWE):** The host system sets this flag to enable or disable the device for remote wake from L1.<br>• 0 - disable<br>• 1 - enable<br>The value of this flag will temporarily (while in L1) override the current setting of the Remote Wake feature set by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, Revision 2.0, Chapter 9. |
| 2:0 | 0x0<br>RW | **L1 Status (L1S):** This field is used by software to determine whether an L1-based suspend request was successful. |

## 51.3.26 Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters necessary for xHC to automatically generate an LPM Token to the downstream device.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 4ACh

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 13:10 | 0x0<br>RW | **Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1.<br>The HIRDD value is is encoded as follows:<br>• 0h: 50 us (default)<br>• 1h: 125 us<br>• 2h: 200 us<br>• ...<br>• Fh: 1.175ms<br>The value of 0h is interpreted as 50 us.<br>Each incrementing value adds 75 us to the previous value.<br><br>**Power Well:** SUS |
| 9:2 | 0x0<br>RW | **L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer.<br>This field shall be set to 00h by assertion of PR to '1'.<br>Following are permissible values:<br>• 00h: 128 us (default)<br>• 01h: 256 us<br>• ...<br>• FFh: 65,280us<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RW | **Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used.<br>Following are permissible values:<br>• 0: Initiate L1 using HIRD only time out (default)<br>• 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD<br>• 2: Reserved<br>• 3: Reserved<br>***Note:*** This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.27 Port Status and Control USB2 (PORTSC4)—Offset 4B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 4B0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Reserved:** Reserved.<br>**Power Well:** SUS |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** Reserved.<br>**Power Well:** Core |
| 29:28 | 0x0<br>RO | **Rsvd2:** Reserved.<br>**Power Well:** Core |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):** Note: This register is sticky.<br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):** Note: This register is sticky.<br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):** Note: This register is sticky.<br>**Power Well:** SUS |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** Reserved.<br>**Power Well:** SUS |
| 23 | 0x0<br>RW/C | **Reserved:** Reserved.<br>**Power Well:** SUS |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):** Note: This register is sticky.<br>**Power Well:** SUS |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** Note: This register is sticky.<br>**Power Well:** SUS |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):** Note: This register is sticky.<br>**Power Well:** SUS |
| 19 | 0x0<br>RW/C | **Reserved:** Reserved.<br>**Power Well:** SUS |
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):** Note: This register is sticky.<br>**Power Well:** SUS |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** Note: This register is sticky.<br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):** Reserved.<br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 9 | 0x1<br>RW | **Port Power (PP):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 8:5 | 0x5<br>RW | **Port Link State (PLS):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Over-current Active (OCA):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):** Note: This register is sticky.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):** Note: This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.28 Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4B4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW | **Port Test Control (PTC):** When this field is '0', the port is not operating in a test mode. (Default) A non-zero value indicates that the port is operating in test mode in indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Disabled state. If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error.<br>The encoding of the Test Mode bits for a USB 2.0 port are:<br>Value Test Mode<br>• 0h - Test mode not enabled<br>• 1h - Test J_STATE<br>• 2h - Test K_STATE<br>• 3h - Test SE0_NAK<br>• 4h - Test Packet<br>• 5h - Test FORCE_ENABLE<br>• 6h-14h - Reserved.<br>• 15 - Port Test Control Error |
| 27:17 | 0x0<br>RO | Reserved. |
| 16 | 0x0<br>RW | **Hardware LPM Enable (HLE):**<br>• 0 - Disable<br>• 1 - Enable. When this bit is a 1, hardware controlled LPM shall be enabled for this port. Refer to Section 4 of the USB 2.0 LPM Specification for more information. |
| 15:8 | 0x0<br>RO | Reserved. |
| 7:4 | 0x0<br>RW | **Best Effort Service Latency:** System software sets this field to indicate the recipient device how long the xHCI will driver resume if it initiates an exit from L1. |
| 3 | 0x0<br>RW | **Remote Wake Enable (RWE):** The host system sets this flag to enable or disable the device for remote wake from L1.<br>• 0 - disable<br>• 1 - enable<br>The value of this flag will temporarily (while in L1) override the current setting of the Remote Wake feature set by the standard Set/ClearFeature() commands defined in Universal Serial Bus Specification, Revision 2.0, Chapter 9. |
| 2:0 | 0x0<br>RW | **L1 Status (L1S):** This field is used by software to determine whether an L1-based suspend request was successful. |

## 51.3.29 Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 4BCh

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 13:10 | 0x0<br>RW | **Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.<br><br>**Power Well:** SUS |
| 9:2 | 0x0<br>RW | **L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RW | **Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.<br><br>**Power Well:** SUS |

## 51.3.30 Port Status and Control USB3 (PORTSC5)—Offset 4C0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 02A0h

**BAR and Offset:** [MBAR] + 4C0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Warm Port Reset (WPR):**<br>When software sets this bit to be "1", the Warm Reset sequence as defined in the USB3 specification is initiated and the PR flag is set to "1". This flag shall always return "0" when read. This flag only applies to USB3 protocol ports. |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** This bit indicates if this port has a removable device.<br>• 0 - Device is removable<br>• 1 - Device is non-removable |
| 29:28 | 0x0<br>RO | Reserved. |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to over-current conditions as system wake-up events. |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device disconnects as system wake-up events. |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device connects as system wake-up events. |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** This bit indicates that far-end terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable to advance to the Enabled state. Software shall clear this bit by writing a 1b to the WPR bit or the xHC shall clear this bit if the CSS bit transitions to 1. |
| 23 | 0x0<br>RW/C | **Port Config Error Change (CEC):** This flag indicates that port failed to configure its link partner.<br>• 0= No change.<br>• 1= Port Config Error Detected.<br>Software shall clear this but by writing a "1" to it. |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>• 0 - No change<br>• 1 - Link Status Change<br>This flag is set to '1' due to the following Port Link (PLS) transitions: |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** This flag is set to '1' due to a '1' to '0' transition of Port Reset (PR). For example, when any reset processing on this port is complete.<br>• 0 - No change<br>• 1 - Reset Complete |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):** The functionality of this bit is not dependent upon the port owner. Software clears this bit by writing a 1 to it.<br>• 0 - No change. (Default)<br>• 1 - There is a change to Overcurrent Active. |
| 19 | 0x0<br>RW/C | **Warm Port Reset Change (WRC):** This bit is set when Warm Reset processing on this port completes.<br>• 0 - No change. (Default)<br>• 1 - Warm reset complete |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>• 0 - No change. (Default)<br>• 1 - There is a change to PED bit. |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** R?WC. This flag indicates a change has occurred in the port's Current Connect Status (CCS) or Cold Attach Status (CAS) bits.<br>• 0 - No change. (Default)<br>• 1 - There is a change to the CCS or CAS bit.<br>The xHC sets this bit to 1b for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the change condition root hub hardware will be "setting" an already-set bit (that is, the bit will remain 1b). |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):**<br>• 0 - When 0b, write data in PLS field is ignored. (Default)<br>• 1 - When this bit is set to 1b on a write references to this register, this flag enables writes to the PLS field.<br>Reads to this bit return '0'. |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):** Note: This register is sticky. |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):** A device attached to this port operates at a speed defined by the following codes:<br>Value Speed<br>• 0001b - full-speed<br>• 0010b - low speed<br>• 0011b - Highspeed<br>• All other values reserved.<br>Refer to the eXtensible Host Controller Interface for Universal Serial Bus Specification for additional details. |
| 9 | 0x1<br>RW | **Port Power (PP):** Read-only with a value of 1. This indicates that the port does have power. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8:5 | 0x5<br>RW | **Port Link State (PLS):** This field is used to power manage the port and reflects its currentlink state. When the port is in the Enabled state, system software may set the link U-state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port.<br><br>Write Value Description<br>• 0 - The link shall transition to a U0 state from any of the U-states.<br>• 2 - USB 2.0 ports only. The link should transition to the U2 State.<br>• 3 - The link shall transition to a U3 state from any of the U-states. This action selectively suspends the device connected to this port. While the Port LinkState = U3 the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port.<br>• 5 - USB 3.0 ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a Rx Detect state and the port shall transition to the Disconnected state, else ignored.<br>• 15 - USB 2.0 ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored.<br>• All other values Ignored<br><br>Read Value Definition<br>• 0 - Link is in the U0 State<br>• 1 - Link is in the U1 State<br>• 2 - Link is in the U2 State<br>• 3 - Link is in the U3 State (Device Suspended)<br>• 4 - Link is in the Disabled State<br>• 5 - Link is in the RxDetect State<br>• 6 - Link is in the Inactive State<br>• 7 - Link is in the Polling State<br>• 8 - Link is in the Recovery State<br>• 9 - Link is in the Hot Reset State<br>• 10 - Link is in the Compliance Mode State<br>• 11 - Link is in the Test Mode State<br>• 12-14 - Reserved<br>• 15 Link is in the Resume State |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** When software writes a 1 to this bit (from a 0), the bus reset sequence as<br>• 1 - port in reset<br>• 0 - port not in reset |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>• 0 - This port does not have an overcurrent condition. (Default)<br>• 1 - This port currently has an overcurrent condition. This bit will automatically transition from 1 to 0 when the over current condition is removed. The PCH automatically disables the port when the overcurrent active bit is 1. |
| 2 | 0x0<br>RO | Reserved. |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):** Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this bit. Ports can be disabled by either a fault condition (disconnect even or other fault condition) or by host software. The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.<br>• 0 - disable<br>• 1 - enable (default) |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):** This value reflects the current state of the port and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>• 0 - no device is present<br>• 1 - device is present on port. |

## 51.3.31 Port Power Management Status and Control USB3 (PORTPMSC5)—Offset 4C4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4C4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **Force Link PM Accept (FLA):** Reserved.<br><br>**Power Well:** SUS |
| 15:8 | 0x0<br>RW/S | **U2 Timeout (U2T):** Reserved.<br><br>**Power Well:** SUS |
| 7:0 | 0x0<br>RW/S | **U1 Timeout (U1T):** Reserved.<br><br>**Power Well:** SUS |

## 51.3.32 USB3 Port Link Info (PORTLI5)—Offset 4C8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4C8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Link Error Count (LEC):** Reserved.<br><br>**Power Well:** Core |

## 51.3.33 Port Status and Control USB3 (PORTSC6)—Offset 4D0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 4D0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Warm Port Reset (WPR):**<br>When software sets this bit to be "1", the Warm Reset sequence as defined in the USB3 specification is initiated and the PR flag is set to "1". This flag shall always return "0" when read. This flag only applies to USB3 protocol ports. |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** This bit indicates if this port has a removable device.<br>• 0 - Device is removable<br>• 1 - Device is non-removable |
| 29:28 | 0x0<br>RO | Reserved. |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to over-current conditions as system wake-up events. |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device disconnects as system wake-up events. |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device connects as system wake-up events. |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** This bit indicates that far-end terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable to advance to the Enabled state. Software shall clear this bit by writing a 1b to the WPR bit or the xHC shall clear this bit if the CSS bit transitions to 1. |
| 23 | 0x0<br>RW/C | **Port Config Error Change (CEC):** This flag indicates that port failed to configure its link partner.<br>• 0= No change.<br>• 1= Port Config Error Detected.<br>Software shall clear this but by writing a "1" to it. |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>• 0 - No change<br>• 1 - Link Status Change<br>This flag is set to '1' due to the following Port Link (PLS) transitions: |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** This flag is set to '1' due to a '1' to '0' transition of Port Reset (PR). For example, when any reset processing on this port is complete.<br>• 0 - No change<br>• 1 - Reset Complete |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):** The functionality of this bit is not dependent upon the port owner. Software clears this bit by writing a 1 to it.<br>• 0 - No change. (Default)<br>• 1 - There is a change to Overcurrent Active. |
| 19 | 0x0<br>RW/C | **Warm Port Reset Change (WRC):** This bit is set when Warm Reset processing on this port completes.<br>• 0 - No change. (Default)<br>• 1 - Warm reset complete |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>• 0 - No change. (Default)<br>• 1 - There is a change to PED bit. |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** R?WC. This flag indicates a change has occurred in the port's Current Connect Status (CCS) or Cold Attach Status (CAS) bits.<br>• 0 - No change. (Default)<br>• 1 - There is a change to the CCS or CAS bit.<br>The xHC sets this bit to 1b for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the change condition root hub hardware will be "setting" an already-set bit (that is, the bit will remain 1b). |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):**<br>• 0 - When 0b, write data in PLS field is ignored. (Default)<br>• 1 - When this bit is set to 1b on a write references to this register, this flag enables writes to the PLS field.<br>Reads to this bit return '0'. |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):** Note: This register is sticky. |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):** A device attached to this port operates at a speed defined by the following codes:<br>Value Speed<br>• 0001b - full-speed<br>• 0010b - low speed<br>• 0011b - Highspeed<br>• All other values reserved.<br>Refer to the eXtensible Host Controller Interface for Universal Serial Bus Specification for additional details. |
| 9 | 0x1<br>RW | **Port Power (PP):** Read-only with a value of 1. This indicates that the port does have power. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8:5 | 0x5<br>RW | **Port Link State (PLS):** This field is used to power manage the port and reflects its currentlink state. When the port is in the Enabled state, system software may set the link U-state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port.<br><br>Write Value Description<br>• 0 - The link shall transition to a U0 state from any of the U-states.<br>• 2 - USB 2.0 ports only. The link should transition to the U2 State.<br>• 3 - The link shall transition to a U3 state from any of the U-states. This action selectively suspends the device connected to this port. While the Port LinkState = U3 the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port.<br>• 5 - USB 3.0 ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a Rx Detect state and the port shall transition to the Disconnected state, else ignored.<br>• 15 - USB 2.0 ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored.<br>• All other values Ignored<br><br>Read Value Definition<br>• 0 - Link is in the U0 State<br>• 1 - Link is in the U1 State<br>• 2 - Link is in the U2 State<br>• 3 - Link is in the U3 State (Device Suspended)<br>• 4 - Link is in the Disabled State<br>• 5 - Link is in the RxDetect State<br>• 6 - Link is in the Inactive State<br>• 7 - Link is in the Polling State<br>• 8 - Link is in the Recovery State<br>• 9 - Link is in the Hot Reset State<br>• 10 - Link is in the Compliance Mode State<br>• 11 - Link is in the Test Mode State<br>• 12-14 - Reserved<br>• 15 Link is in the Resume State |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** When software writes a 1 to this bit (from a 0), the bus reset sequence as<br>• 1 - port in reset<br>• 0 - port not in reset |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>• 0 - This port does not have an overcurrent condition. (Default)<br>• 1 - This port currently has an overcurrent condition. This bit will automatically transition from 1 to 0 when the over current condition is removed. The PCH automatically disables the port when the overcurrent active bit is 1. |
| 2 | 0x0<br>RO | Reserved. |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):** Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this bit. Ports can be disabled by either a fault condition (disconnect even or other fault condition) or by host software. The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.<br>• 0 - disable<br>• 1 - enable (default) |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):** This value reflects the current state of the port and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>• 0 - no device is present<br>• 1 - device is present on port. |

## 51.3.34 Port Power Management Status and Control USB3 (PORTPMSC6)—Offset 4D4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4D4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **Force Link PM Accept (FLA):** Reserved.<br><br>**Power Well:** SUS |
| 15:8 | 0x0<br>RW/S | **U2 Timeout (U2T):** Reserved.<br><br>**Power Well:** SUS |
| 7:0 | 0x0<br>RW/S | **U1 Timeout (U1T):** Reserved.<br><br>**Power Well:** SUS |

## 51.3.35 USB3 Port Link Info (PORTLI6)—Offset 4D8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4D8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Link Error Count (LEC):** Reserved.<br><br>**Power Well:** Core |

## 51.3.36 Port Status and Control USB3 (PORTSC7)—Offset 4E0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 4E0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Warm Port Reset (WPR):**<br>When software sets this bit to be "1", the Warm Reset sequence as defined in the USB3 specification is initiated and the PR flag is set to "1". This flag shall always return "0" when read. This flag only applies to USB3 protocol ports. |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** This bit indicates if this port has a removable device.<br>• 0 - Device is removable<br>• 1 - Device is non-removable |
| 29:28 | 0x0<br>RO | Reserved. |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to over-current conditions as system wake-up events. |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device disconnects as system wake-up events. |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device connects as system wake-up events. |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** This bit indicates that far-end terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable to advance to the Enabled state. Software shall clear this bit by writing a 1b to the WPR bit or the xHC shall clear this bit if the CSS bit transitions to 1. |
| 23 | 0x0<br>RW/C | **Port Config Error Change (CEC):** This flag indicates that port failed to configure its link partner.<br>• 0= No change.<br>• 1= Port Config Error Detected.<br>Software shall clear this but by writing a "1" to it. |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>• 0 - No change<br>• 1 - Link Status Change<br>This flag is set to '1' due to the following Port Link (PLS) transitions: |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** This flag is set to '1' due to a '1' to '0' transition of Port Reset (PR). For example, when any reset processing on this port is complete.<br>• 0 - No change<br>• 1 - Reset Complete |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):** The functionality of this bit is not dependent upon the port owner. Software clears this bit by writing a 1 to it.<br>• 0 - No change. (Default)<br>• 1 - There is a change to Overcurrent Active. |
| 19 | 0x0<br>RW/C | **Warm Port Reset Change (WRC):** This bit is set when Warm Reset processing on this port completes.<br>• 0 - No change. (Default)<br>• 1 - Warm reset complete |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>• 0 - No change. (Default)<br>• 1 - There is a change to PED bit. |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** R?WC. This flag indicates a change has occurred in the port's Current Connect Status (CCS) or Cold Attach Status (CAS) bits.<br>• 0 - No change. (Default)<br>• 1 - There is a change to the CCS or CAS bit.<br>The xHC sets this bit to 1b for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the change condition root hub hardware will be "setting" an already-set bit (that is, the bit will remain 1b). |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):**<br>• 0 - When 0b, write data in PLS field is ignored. (Default)<br>• 1 - When this bit is set to 1b on a write references to this register, this flag enables writes to the PLS field.<br>Reads to this bit return '0'. |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):** Note: This register is sticky. |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):** A device attached to this port operates at a speed defined by the following codes:<br>Value Speed<br>• 0001b - full-speed<br>• 0010b - low speed<br>• 0011b - Highspeed<br>• All other values reserved.<br>Refer to the eXtensible Host Controller Interface for Universal Serial Bus Specification for additional details. |
| 9 | 0x1<br>RW | **Port Power (PP):** Read-only with a value of 1. This indicates that the port does have power. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8:5 | 0x5<br>RW | **Port Link State (PLS):** This field is used to power manage the port and reflects its currentlink state. When the port is in the Enabled state, system software may set the link U-state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port.<br><br>Write Value Description<br>• 0 - The link shall transition to a U0 state from any of the U-states.<br>• 2 - USB 2.0 ports only. The link should transition to the U2 State.<br>• 3 - The link shall transition to a U3 state from any of the U-states. This action selectively suspends the device connected to this port. While the Port LinkState = U3 the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port.<br>• 5 - USB 3.0 ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a Rx Detect state and the port shall transition to the Disconnected state, else ignored.<br>• 15 - USB 2.0 ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored.<br>• All other values Ignored<br><br>Read Value Definition<br>• 0 - Link is in the U0 State<br>• 1 - Link is in the U1 State<br>• 2 - Link is in the U2 State<br>• 3 - Link is in the U3 State (Device Suspended)<br>• 4 - Link is in the Disabled State<br>• 5 - Link is in the RxDetect State<br>• 6 - Link is in the Inactive State<br>• 7 - Link is in the Polling State<br>• 8 - Link is in the Recovery State<br>• 9 - Link is in the Hot Reset State<br>• 10 - Link is in the Compliance Mode State<br>• 11 - Link is in the Test Mode State<br>• 12-14 - Reserved<br>• 15 Link is in the Resume State |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** When software writes a 1 to this bit (from a 0), the bus reset sequence as<br>• 1 - port in reset<br>• 0 - port not in reset |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>• 0 - This port does not have an overcurrent condition. (Default)<br>• 1 - This port currently has an overcurrent condition. This bit will automatically transition from 1 to 0 when the over current condition is removed. The PCH automatically disables the port when the overcurrent active bit is 1. |
| 2 | 0x0<br>RO | Reserved. |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):** Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this bit. Ports can be disabled by either a fault condition (disconnect even or other fault condition) or by host software. The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.<br>• 0 - disable<br>• 1 - enable (default) |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):** This value reflects the current state of the port and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>• 0 - no device is present<br>• 1 - device is present on port. |

## 51.3.37 Port Power Management Status and Control USB3 (PORTPMSC7)—Offset 4E4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4E4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **Force Link PM Accept (FLA):** Reserved.<br><br>**Power Well:** SUS |
| 15:8 | 0x0<br>RW/S | **U2 Timeout (U2T):** Reserved.<br><br>**Power Well:** SUS |
| 7:0 | 0x0<br>RW/S | **U1 Timeout (U1T):** Reserved.<br><br>**Power Well:** SUS |

## 51.3.38 USB3 Port Link Info (PORTLI7)—Offset 4E8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4E8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Link Error Count (LEC):** Reserved.<br><br>**Power Well:** Core |

## 51.3.39 Port Status and Control USB3 (PORTSC8)—Offset 4F0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000002A0h

**BAR and Offset:** [MBAR] + 4F0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1S | **Warm Port Reset (WPR):**<br>When software sets this bit to be "1", the Warm Reset sequence as defined in the USB3 specification is initiated and the PR flag is set to "1". This flag shall always return "0" when read. This flag only applies to USB3 protocol ports. |
| 30 | 0x0<br>RW/L | **Device Removable (DR):** This bit indicates if this port has a removable device.<br>• 0 - Device is removable<br>• 1 - Device is non-removable |
| 29:28 | 0x0<br>RO | Reserved. |
| 27 | 0x0<br>RW | **Wake on Over-current Enable (WOE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to over-current conditions as system wake-up events. |
| 26 | 0x0<br>RW | **Wake on Disconnect Enable (WDE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device disconnects as system wake-up events. |
| 25 | 0x0<br>RW | **Wake on Connect Enable (WCE):**<br>• 0 - Disable. (Default)<br>• 1 - Enable. Writing this but to a 1b enables the port to be sensitive to device connects as system wake-up events. |
| 24 | 0x0<br>RO | **Cold Attach Status (CAS):** This bit indicates that far-end terminations were detected in the Disconnected state and the Root Hub Port State Machine was unable to advance to the Enabled state. Software shall clear this bit by writing a 1b to the WPR bit or the xHC shall clear this bit if the CSS bit transitions to 1. |
| 23 | 0x0<br>RW/C | **Port Config Error Change (CEC):** This flag indicates that port failed to configure its link partner.<br>• 0= No change.<br>• 1= Port Config Error Detected.<br>Software shall clear this but by writing a "1" to it. |
| 22 | 0x0<br>RW/C | **Port Link State Change (PLC):**<br>• 0 - No change<br>• 1 - Link Status Change<br>This flag is set to '1' due to the following Port Link (PLS) transitions: |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** This flag is set to '1' due to a '1' to '0' transition of Port Reset (PR). For example, when any reset processing on this port is complete.<br>• 0 - No change<br>• 1 - Reset Complete |
| 20 | 0x0<br>RW/C | **Over-current Change (OCC):** The functionality of this bit is not dependent upon the port owner. Software clears this bit by writing a 1 to it.<br>• 0 - No change. (Default)<br>• 1 - There is a change to Overcurrent Active. |
| 19 | 0x0<br>RW/C | **Warm Port Reset Change (WRC):** This bit is set when Warm Reset processing on this port completes.<br>• 0 - No change. (Default)<br>• 1 - Warm reset complete |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18 | 0x0<br>RW/C | **Port Enabled Disabled Change (PEC):**<br>• 0 - No change. (Default)<br>• 1 - There is a change to PED bit. |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** R?WC. This flag indicates a change has occurred in the port's Current Connect Status (CCS) or Cold Attach Status (CAS) bits.<br>• 0 - No change. (Default)<br>• 1 - There is a change to the CCS or CAS bit.<br>The xHC sets this bit to 1b for all changes to the port device connect status, even if system software has not cleared an existing Connect Status Change. For example, the insertion status changes twice before system software has cleared the change condition root hub hardware will be "setting" an already-set bit (that is, the bit will remain 1b). |
| 16 | 0x0<br>RW | **Port Link State Write Strobe (LWS):**<br>• 0 - When 0b, write data in PLS field is ignored. (Default)<br>• 1 - When this bit is set to 1b on a write references to this register, this flag enables writes to the PLS field.<br>Reads to this bit return '0'. |
| 15:14 | 0x0<br>RW | **Port Indicator Control (PIC):** Note: This register is sticky. |
| 13:10 | 0x0<br>RW | **Port Speed (PortSpeed):** A device attached to this port operates at a speed defined by the following codes:<br>Value Speed<br>• 0001b - full-speed<br>• 0010b - low speed<br>• 0011b - Highspeed<br>• All other values reserved.<br>Refer to the eXtensible Host Controller Interface for Universal Serial Bus Specification for additional details. |
| 9 | 0x1<br>RW | **Port Power (PP):** Read-only with a value of 1. This indicates that the port does have power. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8:5 | 0x5<br>RW | **Port Link State (PLS):** This field is used to power manage the port and reflects its currentlink state. When the port is in the Enabled state, system software may set the link U-state by writing this field. System software may also write this field to force a Disabled to Disconnected state transition of the port.<br><br>Write Value Description<br>• 0 - The link shall transition to a U0 state from any of the U-states.<br>• 2 - USB 2.0 ports only. The link should transition to the U2 State.<br>• 3 - The link shall transition to a U3 state from any of the U-states. This action selectively suspends the device connected to this port. While the Port LinkState = U3 the hub does not propagate downstream-directed traffic to this port, but the hub will respond to resume signaling from the port.<br>• 5 - USB 3.0 ports only. If the port is in the Disabled state (PLS = Disabled, PP = 1), then the link shall transition to a Rx Detect state and the port shall transition to the Disconnected state, else ignored.<br>• 15 - USB 2.0 ports only. If the port is in the U3 state (PLS = U3), then the link shall remain in the U3 state and the port shall transition to the U3Exit substate, else ignored.<br>• All other values Ignored<br><br>Read Value Definition<br>• 0 - Link is in the U0 State<br>• 1 - Link is in the U1 State<br>• 2 - Link is in the U2 State<br>• 3 - Link is in the U3 State (Device Suspended)<br>• 4 - Link is in the Disabled State<br>• 5 - Link is in the RxDetect State<br>• 6 - Link is in the Inactive State<br>• 7 - Link is in the Polling State<br>• 8 - Link is in the Recovery State<br>• 9 - Link is in the Hot Reset State<br>• 10 - Link is in the Compliance Mode State<br>• 11 - Link is in the Test Mode State<br>• 12-14 - Reserved<br>• 15 Link is in the Resume State |
| 4 | 0x0<br>RW/1S | **Port Reset (PR):** When software writes a 1 to this bit (from a 0), the bus reset sequence as<br>• 1 - port in reset<br>• 0 - port not in reset |
| 3 | 0x0<br>RW | **Over-current Active (OCA):**<br>• 0 - This port does not have an overcurrent condition. (Default)<br>• 1 - This port currently has an overcurrent condition. This bit will automatically transition from 1 to 0 when the over current condition is removed. The PCH automatically disables the port when the overcurrent active bit is 1. |
| 2 | 0x0<br>RO | Reserved. |
| 1 | 0x0<br>RW/C | **Port Enabled Disabled (PED):** Ports can only be enabled by the host controller as a part of the reset and enable. Software cannot enable a port by writing a 1 to this bit. Ports can be disabled by either a fault condition (disconnect even or other fault condition) or by host software. The bit status does not change until the port state actually changes. There may be a delay in disabling or enabling a port due to other host controller and bus events.<br>• 0 - disable<br>• 1 - enable (default) |
| 0 | 0x0<br>RW | **Current Connect Status (CCS):** This value reflects the current state of the port and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set.<br>• 0 - no device is present<br>• 1 - device is present on port. |

## 51.3.40 Port Power Management Status and Control USB3 (PORTPMSC8)—Offset 4F4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4F4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **Force Link PM Accept (FLA):** Reserved.<br><br>**Power Well:** SUS |
| 15:8 | 0x0<br>RW/S | **U2 Timeout (U2T):** Reserved.<br><br>**Power Well:** SUS |
| 7:0 | 0x0<br>RW/S | **U1 Timeout (U1T):** Reserved.<br><br>**Power Well:** SUS |

## 51.3.41 USB3 Port Link Info (PORTLI8)—Offset 4F8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 4F8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Link Error Count (LEC):** Reserved.<br><br>**Power Well:** Core |

## 51.3.42 Microframe Index (RTMFINDEX)—Offset 2000h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 2000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 13:0 | 0x0<br>RO | **Microframe Index (IMAN0):** Reserved.<br><br>**Power Well:** Core |

## 51.3.43 Interrupter 1 Management (IMAN0)—Offset 2020h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2020h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br>**Power Well:** Core |

## 51.3.44 Interrupter 1 Moderation (IMOD0)—Offset 2024h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2024h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br>**Power Well:** Core |

## 51.3.45 Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2028h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br>**Power Well:** Core |

## 51.3.46 Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2030h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.47 Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2034h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.48 Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 2038h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.49 Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 203Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.50 Interrupter 2 Management (IMAN1)—Offset 2040h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2040h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.51 Interrupter 2 Moderation (IMOD1)—Offset 2044h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2044h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.52 Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2048h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.53 Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2050h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.54 Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2054h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.55 Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2058h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.56 Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 205Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.57 Interrupter 3 Management (IMAN2)—Offset 2060h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2060h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.58 Interrupter 3 Moderation (IMOD2)—Offset 2064h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2064h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.59 Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2068h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.60 Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2070h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.61 Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2074h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.62 Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2078h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.63 Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 207Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.64 Interrupter 4 Management (IMAN3)—Offset 2080h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2080h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.65 Interrupter 4 Moderation (IMOD3)—Offset 2084h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2084h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.66 Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2088h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.67 Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2090h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.68 Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 2094h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.69 Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 2098h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.70 Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 209Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.71 Interrupter 5 Management (IMAN4)—Offset 20A0h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20A0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br>**Power Well:** Core |

## 51.3.72 Interrupter 5 Moderation (IMOD4)—Offset 20A4h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20A4h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br>**Power Well:** Core |

## 51.3.73 Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20A8h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br>**Power Well:** Core |

## 51.3.74 Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20B0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.75 Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20B4h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.76 Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20B8h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.77 Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 20BCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.78 Interrupter 6 Management (IMAN5)—Offset 20C0h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 20C0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.79 Interrupter 6 Moderation (IMOD5)—Offset 20C4h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000FA0h

**BAR and Offset:** [MBAR] + 20C4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.80 Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20C8h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.81 Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20D0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.82 Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20D4h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.83 Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20D8h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.84 Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20DCh

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.85 Interrupter 7 Management (IMAN6)—Offset 20E0h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20E0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.86 Interrupter 7 Moderation (IMOD6)—Offset 20E4h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20E4h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.87 Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20E8h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.88 Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 20F0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.89 Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 20F4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.90 Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 20F8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.91 Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 20FCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.92 Interrupter 8 Management (IMAN7)—Offset 2100h

There are 8 IMAN registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2100h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Interrupt Enable (IE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/C | **Interrupt Pending (IP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.93 Interrupter 8 Moderation (IMOD7)—Offset 2104h

There are 8 IMOD registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2104h

**Default:** 00000FA0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Interrupt Moderation Counter (IMODC):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0xfa0<br>RW | **Interrupt Moderation Interval (IMODI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.94 Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2108h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** Reserved.<br><br>**Power Well:** Core |

## 51.3.95 Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2110h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBA):** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |

## 51.3.96 Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2114h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Segment Table Base Address (ERSTBA):** Reserved.<br><br>**Power Well:** Core |

## 51.3.97 Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 2118h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/C | **Event Handler Busy (EHB):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** Reserved.<br><br>**Power Well:** Core |

## 51.3.98 Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 211Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Event Ring Dequeue Pointer (ERDP):** Reserved.<br><br>**Power Well:** Core |

## 51.3.99 Door Bell 1 (DB0)—Offset 3000h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3000h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.100 Door Bell 2 (DB1)—Offset 3004h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3004h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.101 Door Bell 3 (DB2)—Offset 3008h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3008h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.102 Door Bell 4 (DB3)—Offset 300Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 300Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.103 Door Bell 5 (DB4)—Offset 3010h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3010h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.104 Door Bell 6 (DB5)—Offset 3014h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3014h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.105 Door Bell 7 (DB6)—Offset 3018h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3018h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.106 Door Bell 8 (DB7)—Offset 301Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 301Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.107 Door Bell 9 (DB8)—Offset 3020h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3020h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.108 Door Bell 10 (DB9)—Offset 3024h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3024h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.109 Door Bell 11 (DB10)—Offset 3028h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3028h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.110 Door Bell 12 (DB11)—Offset 302Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 302Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.111 Door Bell 13 (DB12)—Offset 3030h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3030h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.112 Door Bell 14 (DB13)—Offset 3034h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3034h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.113 Door Bell 15 (DB14)—Offset 3038h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3038h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.114 Door Bell 16 (DB15)—Offset 303Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 303Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.115 Door Bell 17 (DB16)—Offset 3040h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3040h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.116 Door Bell 18 (DB17)—Offset 3044h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3044h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.117 Door Bell 19 (DB18)—Offset 3048h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3048h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.118 Door Bell 20 (DB19)—Offset 304Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 304Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.119 Door Bell 21 (DB20)—Offset 3050h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3050h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.120 Door Bell 22 (DB21)—Offset 3054h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3054h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.121 Door Bell 23 (DB22)—Offset 3058h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3058h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.122 Door Bell 24 (DB23)—Offset 305Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 305Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.123 Door Bell 25 (DB24)—Offset 3060h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3060h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.124 Door Bell 26 (DB25)—Offset 3064h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3064h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.125 Door Bell 27 (DB26)—Offset 3068h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3068h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.126 Door Bell 28 (DB27)—Offset 306Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 306Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.127 Door Bell 29 (DB28)—Offset 3070h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3070h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.128 Door Bell 30 (DB29)—Offset 3074h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3074h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.129 Door Bell 31 (DB30)—Offset 3078h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3078h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.130 Door Bell 32 (DB31)—Offset 307Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 307Ch

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.131 Door Bell 32 (DB32)—Offset 3080h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 3080h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **DB Stream ID (DSI):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DB Target (DT):** This field defines the target of the doorbell reference.<br>Doorbell register 0: Dedicated to Command Ring and decodes this field differently than other doorbell registers.<br><br>**Power Well:** Core |

## 51.3.132 XECP_SUPP_USB2_1—Offset 8004h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 20425355h

**BAR and Offset:** [MBAR] + 8004h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x20425355<br>RO | **XECP_SUPP_USB2_1:** Namestring USB<br><br>**Power Well:** Core |

## 51.3.133 XECP_SUPP_USB2_2—Offset 8008h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 30190401h

**BAR and Offset:** [MBAR] + 8008h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x3<br>RO | **Protocol Speed ID Count (PROT_SPD_ID_CNT):** 3 USB 2.0 Speed (High, Full, Low)<br><br>**Power Well:** Core |
| 27:21 | 0x0<br>RO | **Rsvd0:** Reserved.<br><br>**Power Well:** Core |
| 20 | 0x1<br>RW/L | **BESL LPM Capability (BLC):** Bit is set to 1 to indicate that the the ports described by this xHCI Supported Protocol Capability will apply BESL timing to BESL and BESLD fields of the PORTPMSC and PORTHLPMCC registers.<br><br>**Power Well:** Core |
| 19 | 0x1<br>RW/L | **Protocol Defined - Hardware LMP Capability (HLC):** Reserved.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RO | **Protocol Defined - Integrated Hub Implementation (IHI):** Reserved.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RO | **Protocol Defined - High SPeed Only (HSO):** Reserved.<br><br>**Power Well:** Core |
| 16 | 0x1<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x4<br>RO | **Compatible Port Count (CPC):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x1<br>RO | **Compatible Port Offset (CPO):** Reserved.<br><br>**Power Well:** Core |

## 51.3.134 XECP_SUPP_USB3_1—Offset 8024h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 20425355h

**BAR and Offset:** [MBAR] + 8024h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x20425355<br>RO | **XECP_SUPP_USB2_1:** Namestring USB<br><br>**Power Well:** Core |

## 51.3.135 XECP_SUPP_USB3_2—Offset 8028h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 10000405h

**BAR and Offset:** [MBAR] + 8028h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x1<br>RO | **Protocol Speed ID Count (PROT_SPD_ID_CNT):** 1 USB 3.0 Speed (Supper Speed)<br><br>**Power Well:** Core |
| 27:16 | 0x0<br>RO | **Rsvd0:** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x4<br>RO | **Compatible Port Count (CPC):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x5<br>RO | **Compatible Port Offset (CPO):** Reserved.<br><br>**Power Well:** Core |

## 51.3.136 Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h

This is a register that describe the host controller the extended cap location. It includes the , XECP_HOST_NEXT_CAP_OFFSET and VEND_DEF_HOST_CAP_ID_192. This register is not subject to HW save and restore.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 004DFFC0h

**BAR and Offset:** [MBAR] + 8070h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 23:16 | 0x4d<br>RW/L | **Valid Length (VALID_LENGTH):** Indicates the number of valid DWords in the capability, that need to be saved and restored This value needs to be updated as new chicken bit registers are added. 47h -) 818Bh is the last valid byte<br><br>**Power Well:** Core |
| 15:8 | 0xff<br>RW/L | **Next Capability Pointer (XECP_HOST_NEXT_CAP_OFFSET):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0xc0<br>RW/L | **Supported Protocol ID (VEND_DEF_HOST_CAP_ID):** Reserved.<br><br>**Power Well:** Core |

## 51.3.137 Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8078h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 9:5 | 0x0<br>WO | **Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots<br><br>**Power Well:** Core |
| 4:1 | 0x0<br>WO | **EP Number (EP_NUM):** 4bits of EP number<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Clear Internal Scheduler's Mask (CISM):** This is a register that is used to clear the internal scheduler's mask that is used to stop scheduling a particular EP. Bit0 indicates the direction of the EP<br><br>**Power Well:** Core |

## 51.3.138 Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 807Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG):** This register is used to clear the internal valid IN EP array that TRM stored in order to guarantee one IN EP per port. This register allows software to clear the valid bit of each port IN EP. This field indicates the port number. For a 2port configuration, only bit1:0 are valid. It can scale for the max number of ports that we support.<br><br>**Power Well:** Core |

## 51.3.139 Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8080h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 9:5 | 0x0<br>WO | **Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots<br><br>**Power Well:** Core |
| 4:1 | 0x0<br>WO | **EP Number (EP_NUM):** 4bits of EP number<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Clear Internal Scheduler's Poll Mask (CISPM):** This is a register that is used to clear the internal scheduler's poll mask that is used to indicate whether we need to poll this EP. This is used for USB2. Bit0 indicates the direction of the EP<br><br>**Power Well:** Core |

## 51.3.140 Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00008100h

**BAR and Offset:** [MBAR] + 8094h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RW | **Scheduler Host Control Reg Cont (SCHED_HOST_CTRL_CONT):** method of USB2 port periodic done check (off by default)<br><br>**Power Well:** Core |
| 20:13 | 0x4<br>RW | **TTE Host Control (TTE_HOST_CTRL):** (0): disable interrupt complete split limit to 3 microframes (1): disable checking of missed microframes (2): disable split error request w/NULL pointer on speculative INs with data payload and no TRB. (3): disable deferred split error request on speculative IN with data payload and no TRB. (7:4): reserved<br><br>**Power Well:** Core |
| 12:11 | 0x0<br>RW | **Cache Size Control Reg (CACHE_SZ_CTRL):** 0: 64 1: 32 2,3: 16<br><br>**Power Well:** Core |
| 10:9 | 0x0<br>RW | **Maximum EP Per Slot (MAX_EP_SLOT):** 0: 32 1: 16 2: 8 3: 4<br><br>**Power Well:** Core |
| 8 | 0x1<br>RW | **Turn on scratch_pad_en (TO_SCRATCH_PAD_EN):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Scheduler Host Control Reg (SCHED_HOST_CTRL):** (0): disable poll delay (1): disable TRM active in EP valid check (2): enable TTE overlap prevention on interrupt IN EPs (at cost of possible service interval slip) (3) enable TTE overlap prevention on interrupt OUT EPs (at cost of possible service interval slip) (5:4) scheduler sort pattern 00 (default) search ISO ahead of interrupt within each service interval 01 - search USB2-ISO, USB3-ISO, USB2-Interrupt, USB3-Interrupt within each service interval 10 - search strictly by interval 11 - search all ISO intervals ahead interrupt intervals and within each interval, USB2 ahead of USB3 (6): disable 1 pack scheduling limit when ISO pending in present microframe (7): enable check to stop scheduling on port that are not connected<br><br>**Power Well:** Core |

## 51.3.141 Power Management Control (PMCTRL_REG)—Offset 80A4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 002DFF90h

**BAR and Offset:** [MBAR] + 80A4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Async PME Source Enable (ASYNC_PME_SRC_EN):** This field allows the async PME source to be allowed to generate PME. This is specifically required for SOCs that do not allow for any clock other than RTC to be available during RTD3.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW | **Legacy PME Source Enable (LEGACY_PME_SRC_EN):** This field allows the legacy PME source to be used in PME generation. The legacy source in in reference to the source prior to the RTD3 changes.<br><br>**Power Well:** SUS |
| 29 | 0x0<br>RW | **Reset Warn Power Gate Trigger Disable (RESET_WARN_PWR_GATE_TRIGGER_DISABLE):** This field controls the actions taken for due to reset warn. 0 - Reset Warn will trigger a HW autonomous Power Gate 1 - Reset Warn will not trigger a HW autonomous Power Gate<br><br>**Power Well:** SUS |
| 28 | 0x0<br>RW | **CLR_PME_FLAG_PULSE_AUX_CCLK:** Reserved.<br><br>**Power Well:** SUS |
| 27 | 0x0<br>RW | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **XLFPSCOUNTSRC:** XLFPSCOUNTSRC (Source for LFPS OFF Counter) 0: Central RTC Counter for LFPS detection 1: Local Counter for LFPS detection<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **XELFPSRTC:** XELFPSRTC (Enable LFPS Filtering on RTC) 0: Use Oscillator clock for LFPS Filtering during P3 1: Use RTC Clock for LFPS Filtering during P3<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RW | **XMPHYSPGDD0I2:** XMPHYSPGDD0I2 (ModPhy Sus Well Power Gate Disable for D0I2) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW | **XMPHYSPGDD0I3:** XMPHYSPGDD0I3 (ModPhy Sus Well Power Gate Disable for D0I3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW | **XMPHYSPGDRTD3:** XMPHYSPGDRTD3 (ModPhy Sus Well Power Gate Disable for RTD3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled<br><br>**Power Well:** SUS |
| 21:18 | 0xb<br>RW | **XD3RTCPTTM:** XD3RTCPTTM (D3 RTC Port Timer Tick Multiplier) This register will be the multiplication factor for determining SSIC Wake Detection Frequency and RXDET based on the XD3RTCPTTC value. If XD3RTCPTTC is 9h and this register is Bh, frequency for RXDET and SSIC H8EXIT detection while MODPHY SUS Power gating is enabled would be 99ms.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **U3 LFPS Periodic Sampling ON Time Control (U3_LFPS_PRDC_SAMPLING_ON_TIME_CTRL):** This field controls the ON time for the LFPS periodic sampling for USB3/SSIC ports. 0 ON time is 2 rtc clocks 1 ON time is 3 rtc clocks Note: This field is ignored if USB3/SSIC PHY SUS Well Power Gating is enabled.<br><br>**Power Well:** SUS |
| 16 | 0x1<br>RW | **AON LFPS Detector Enable Mode (AON_LFPS_DETECTOR_EN_MODE):** 1 - Allow the LFSP Detector in AON to own LFPS detection when the port is in PS3 for U2/U3 - not RxD regardless of port ownership. 0 - Allow the LFPS Detector in AON to own the LFPS detection only when the AON owns the port and in U2/U3 - not RxD<br><br>**Power Well:** SUS |
| 15:8 | 0xff<br>RW | **SS U3 LFPS Detection Threshold (SS_U3_LFPS_DETECTION_THRESHOLD):** This field controls the threshold used to determine when a valid U3 Wake is detected through when using the unfiltered LFPS source. The value on this field will reflect the binary count required to have been detected on the counter being clocked by the unfiltered lfps source to result in a valid U3 wake detection.<br><br>**Power Well:** SUS |
| 7:4 | 0x9<br>RW | **SS_U3_LFPS_PRDC_SAMPLING_OFFTIME_CTRL:** This field controls the OFF time for the LFPS periodic sampling for USB3 Ports 0x0 periodic sampling is disabled. 0x1 OFF time is 1ms 0x2 OFF time is 2ms 0xF OFF time is 15ms The ON Time is determined by the amount of time required to reliably determine if there is a valid LFPS and is HW implementation specific. A speed up mode shall be implemented where this field is in units of us. i.e. 0x1 = 1 us OFF time, 0x2 = 2 us OFF time, etc.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **PS3 LFPS Source Select (PS3_LFPS_SRC_SEL):** 0 LFPS Source is unfiltered 1 LFPS Source is filtered (Rx-Elec-Idle) LFPS Source is Rx-Elec-Idle for any non PS3 state.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **XHCI Engine Autonomous Power Gate Exit Reset Policy (XHC_AUTO_PWRGATE_EXITRST_POLICY):** Controls when the xHCI engine is brought out of reset due to a power ungate. 0 Engine is brought out of reset when D3 to D0 is triggered. This allows for a quick power up sequence while leaving the virtual PCIe LTSSM in L23 is power ungate is not due to D3 to D0. 1 Engine is brought out of reset along with the rest of the IP. This is required for PMC save/restore flow.<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **USB2 Port Wake Unit Coupling Policy (USB2_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB2 Port Wake Units to initiate Port Level Power Off Preparation. 0 RTD3 triggered 1 - Port Triggered when in L1, L2 or Disabled, Disconnected<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **USB3 Port Wake Unit Coupling Policy (USB3_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB3 Port Wake Units to initiate Port Level Power Off Preparation. 0 - RTD3 Triggered 1 - Port Triggered when in PS3 due to RxDetect, U3, U2 or Disabled<br><br>**Power Well:** SUS |

## 51.3.142 PGCB Control (PGCBCTRL_REG)—Offset 80A8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00315555h

**BAR and Offset:** [MBAR] + 80A8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 27:25 | 0x0<br>RW | **IP_INACCESSIBLE_HYSTERESIS_TIMER:**<br>• 000 =50us<br>• 001 =100us<br>• 010 =150us<br>• 011 =200us<br>• 000 =250us<br>• 001 =300us<br>• 010 =350us<br>• 111 =400us<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RW | **Override_Disable (OVERRIDE_DISABLE):** Reserved.<br><br>**Power Well:** SUS |
| 23:20 | 0x3<br>RW | **CDH Aggregation Minimum Wait Time (CDH_MIN_WAIT_TIME):** This field controls the minimum time that we must wait for resets to propagate before allowing Power Up flow to complete. The Power Up flow requires special handling of clocks, reset and sleep signaling which requires the CDH to monitor reset propagation.<br>This timer allow for flexibility on when we consider all resets propagated.<br>• 0h Disabled<br>• 1h 2 clocks<br>• 2h 4 clocks<br>• 3h 8 clocks<br>• 4h 16 clocks<br>• 5h 32 clocks<br>• 6h 64 clocks<br>• 7h 128 clocks<br><br>**Power Well:** SUS |
| 19 | 0x0<br>RW | **RESERVED (CDH_RST_PROPAGATION_CNTRL):** This bit will indicate the mode of operation for the CDH Reset Propagation Aggregator.<br>It allows for a mode where the aggregation will wait for an indication from each CDH before proceeding OR ignore the CDH indication before proceeding.<br>Regardless of the mode, the aggregator will enforce the CDH Aggregation Mini Wait Time. 0 Aggregate all the CDH indications before triggering the CDH Aggregation Mini Wait Time. Once the Min Wait Time is met a notification to the Power Sequencer/Control will be initiated 1 Trigger the CDH Aggregation Mini Wait Time w/o waiting for CDH indication. Once the Min Wait time is met a notification to the Power Sequencer/Control will be initiated.<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW | **MMP_PFET_REQ_OVRD:** This bit will disable the MMP PFET request condition for PGCB control.<br>• 1 MMP PFET Request Ignored<br>• 0 Default<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17:16 | 0x1<br>RW | **PGCB Clock Gate Request to PGCB Sleep (PGCB_CLK_GATE_REQ_2_PGCB_SLEEP):** Value representing the minimum number of delay clocks required between the assertion of pgcb_ip_clkgate_b to the deassertion of pgcb_sleep on PG exit.<br>(This is only applicable for IP-Accessible PG with state-retention enabled.)<br>Please see the description of cfg_tsleepact for more details.<br><br>**Power Well:** SUS |
| 15:14 | 0x1<br>RW | **PGCB Sleep Deassertion to PGCB ISM Unlock Req (PGCB_SLEEP_DEASRTN_2_ISM_UNLOCK_REQ):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of gcb_sleepto the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled):<br>Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.<br><br>**Power Well:** SUS |
| 13:12 | 0x1<br>RW | **PGCB Prim Reset Deassertion to PGCB Next State (PGCB_PRIMRST_DEASRTN_2_NSTATE):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled):<br>Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.<br><br>**Power Well:** SUS |
| 11:10 | 0x1<br>RW | **PGCB Side Reset Deassertion to PGCB Prim Reset Deassertion (PGCB_SIDERST_DEASRTN_2_PRIMERST_DEASRTN):** Value representing the number of delay clocks required between the deassertion of gcb_force_rst_bto the deassertion of gcb_force_prim_rst_b (This is only applicable for IP-Accessible PG.) Please see the description of cfg_tsleepact for more details.<br><br>**Power Well:** SUS |
| 9:8 | 0x1<br>RW | **PGCB Latch Isolation High to PGCB Clock Ungate Request (PGCB_LATCH_ISO_HI_2_PGCB_CLK_UNGATE_REQ):** Value representing the number of delay clocks required between the assertion of pgcb_isol_latchen to the deassertion of pgcb_ip_clkgate_req_b. Please see the description of cfg_tsleepact for more details.<br><br>**Power Well:** SUS |
| 7:6 | 0x1<br>RW | **PGCB Isolation Deassertion to PGCB Latch Isolation High Count (PGCB_ISO_DEASRTN_2_PGCB_ISO_HI_CNT):** Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_en_b to the assertion of gcb_isol_latchen.<br>Please see the description of cfg_tsleepact for more details.<br><br>**Power Well:** SUS |
| 5:4 | 0x1<br>RW | **PGCB Reset Assertion to PGCB Power Down Request Count (PGCB_RST_2_PGCB_PWRDOWN_REQCNT):** For IP-Accessible PG (with state retention):<br>Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_pmc_pg_req_b. For IP-Inaccessible PG (or with state-retention disabled):<br>Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_sleep.<br>Please see the description of cfg_tsleepact for more details.<br><br>**Power Well:** SUS |
| 3:2 | 0x1<br>RW | **PGCB Isolation Assertion to PGCB Reset Assertion Count (PGCB_ISO_ASRTN_2_PGCB_RST_ASRTN_CNT):** Value representing the minimum number of delay clocks required between the assertion of gcb_isol_en_bto the assertion of gcb_force_prim_rst_band gcb_force_rst_b Please see the description of [1:0] for more details.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1:0 | 0x1<br>RW | **PGCB Sleep Assertion to PGCB Isolation Enable Count (PGCB_SLEEP_ASRTN_2_PGCB_ISO_EN_CNT):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the assertion of pgcb_sleep (and the deassertion of pgcb_isol_latchen) to the assertion of pgcb_isol_en_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_latchen to the assertion of pgcb_isol_en_b, as well as the minimum number of delay clocks required between the assertion of pgcb_sleep to the assertion of pgcb_pmc_pg_req_b.<br>Common for all cfg_t* inputs:<br>0 1 clock<br>1 2 clocks<br>0 8 clocks<br>1 256 clocks<br>For any of the cfg_t* inputs that an IP may feel are a don-care and the IP does not feel they will be useful for post-silicon debug, the recommendation is to tie the input to 201 (2 clocks).<br>Otherwise, the inputs may be tied to another relevant value or connected to a configuration register.<br>***Note:*** These signals may only change while pgcb_pwrupidle is asserted, if pgcb_pwrupidle is deasserted it should be valid and stable. (It may change the same cycle that ip_pgcb_pg_rdy_req_b asserts.)<br><br>**Power Well:** SUS |

## 51.3.143 D0I3 Control (DOI3CTRL_REG)—Offset 80ACh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000008h

**BAR and Offset:** [MBAR] + 80ACh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x1<br>RW/1C | **RestoreRequired (RESTORE_REQUIRED):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **D0I3:** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RO | **RESERVED (RSVD1):** Reserved.<br><br>**Power Well:** SUS |

## 51.3.144 HOST_CTRL_MISC_REG—Offset 80B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0001037Fh

**BAR and Offset:** [MBAR] + 80B0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **USB2_LTRUPDT_DIS:** Reserved.<br><br>**Power Well:** Core |
| 30 | 0x0<br>RW | **USB2 Line State Debounce During Port Reset Policy (USB2_LINE_STATE_DEBOUNCE_DURING_PORT_RESET_POLICY):** This register controls how the debounce is enforced during the Port Reset phase. 0 do not enable the line state debounce during port reset. 1 enable the line state debounce during port reset.<br><br>**Power Well:** Core |
| 29 | 0x0<br>RW | **TTE PEXE Credit Fix Disable (TTE_PEXE_CREDIT_FIX_DISABLE):** When set, it disables a fix implemented to re-deem PEXE credits when a port is disconnected<br><br>**Power Well:** Core |
| 28 | 0x0<br>RW | **TTE Scheduling policy (TTE_SCHEDULING_POLICY):** This register controls a fix made to prevent over-scheduling by not account for 188B in each uFrame. Setting this bit will disable the fix and allow for over-scheduling.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **USB3 ITP Delta Timer Source Select (USB3_ITP_DELTA_TIMER_SOURCE_SELECT):** This register selects the source for the delta timer tracking used for ITP generation. 0 the source is a 16.666 ns tick generated from a crystal reference clock 1 - the source is a 16.666 ns tick generated from the aux_cclk. This field needs to remain in sync with Frame Timer Source Select to ensure the are both set or both cleared. There is no support for any other combination.<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW | **Frame Timer Source Select (FRAME_TIMER_SOURCE_SELECT):** This register controls the source for the frame timer. 0 the source for the frame timer is a crystal reference clock 1 the source for the frame timer is the aux_cclk.<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **uFrame Masking Enable (UFRAME_MASKING_ENABLE):** If set, enables the uFrame tick to be masked due to ports being in U3/NC. This controls a fix made to disable the auto masking of uFrame tick due to port state w/o any pipeline idle condition. When cleared, we rely on gating of frame timer due to proper port state and idleness tracking from the pipeline.<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **Late FID Check Disable (LATE_FID_CHECK_DISABLE):** This register disables the Late FID Check performed when starting an ISOCH stream.<br><br>**Power Well:** Core |
| 23:20 | 0x0<br>RW | **RESERVED (RSVD1):** Reserved.<br><br>**Power Well:** Core |
| 19 | 0x0<br>RW | **USB2 Resume Cx Inhibit Disable (USB2_RESUME_CX_INHIBIT_DISABLE):** Controls if USB2 L1 Resume is allowed to contribute to DMA Active which will inhibit Cx state. 0 USB2 L1 Resume is allowed to inhibit Cx via DMA Active 1 USB2 L1 Resume is NOT allowed to inhibit Cx via DMA Active When cleared, Cx will only be inhibited when the DMA traffic for the port begins.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18:16 | 0x1<br>RW | **Extra uFrame (EXTRA_UFRAME):** This register controls the extra number of uFrames added onto the advancing of late FID check.<br><br>**Power Well:** Core |
| 15:0 | 0x37f<br>RW | **Valid Isoch Scheduling Range (VALID_ISOCH_SCHEDULING_RANGE):** This register defines the window in miliseconds from the current Frame that will be considered for scheduling in an upcoming Frame. Anything scheduled outside of this window will be considered as late and will trigger the Missed Service Error.<br><br>**Power Well:** Core |

## 51.3.145 reg_SSPITPE_type (SSPITPE)—Offset 80BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0000000Fh

**BAR and Offset:** [MBAR] + 80BCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 3:0 | 0xf<br>RW | **ITP Transmit Enable (ITP_TRANSMIT_EN):** Width is scaled with USB3 Port Count PortCount=4<br><br>**Power Well:** Core |

## 51.3.146 AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 015FC0F0h

**BAR and Offset:** [MBAR] + 80C0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Reset USB3 AUX/Main PD Logic (RST_U3_AM_PDL):** A reset that is designed to reset all USB3 port AUX + Main power domain logic. This is a debug function. It is called cold reset. Write 1 to this register will issue a cold reset to USB3 ports.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 29 | 0x0<br>RW | **Enable PCIe PIPE Reset As PCIe PHY Reset (EN_PPIPE_PPHY_RST):** This bit set to 1 selects the PCIe PIPE reset as a PCIe PHY power on reset. 0 selects the AUX powerup reset as a PCIe PHY powerup reset.<br><br>**Power Well:** SUS |
| 28 | 0x0<br>RW | **Ensure PIPE Powerdown On PERST# (EN_PPWRDN_PERST):** This bit ensures any PERST# being asserted will cause PIPE powerdown state transition to P1.<br><br>**Power Well:** SUS |
| 27 | 0x0<br>RW | **Disable PERST# Main RST (DIS_PERST_MRST):** There is a feature where we only allow PERST# to be treated as a main powerdown reset when it is asserted during the state that is not in D3. This bit disables this feature when it is set to 1. If this bit is set to 1, it means that any PERST# will be treated as a main power domain reset regardless of its Dstate.<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Disable PCIe PERST# Host Controller (DIS_PERST_HC):** There is a feature where we can disable PCIe PERST# from reset the host controller until power management control module has done its clock switching. This is for a case where PM entered L23 and immediately PERST# asserted. This bit is designed to enable this feature when set to 1.<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Enable Fast Simulation Reset Mode (EN_FAST_RST):** Enable a fast simulation mode for reset function. This is a feature for GLS<br><br>**Power Well:** SUS |
| 24 | 0x1<br>RW | **Prevent USB3 Link Down Reset (PREV_U3_LDRST):** When set to 1 prevent a reset being generated due to the USB3 port link down condition.<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW | **Ignore Main Power Up Reset (IGN_MPU_RST):** When set to 1 ignore the main power up reset for USB3 PIPE PHY reset.<br><br>**Power Well:** SUS |
| 22 | 0x1<br>RW | **RESERVED (RSVD_1):** Reserved.<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW | **Ignore HC Reset USB2 (IGN_HC_RST_U2):** When set to '1' ignore HC reset to reset the USB2 Port logic<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 20 | 0x1<br>RW | **Ignore HC Reset USB PHY (IGN_HC_RST_UP_POR):** When set to '1' ignore HC reset to the USB PHY power-on reset<br><br>**Power Well:** SUS |
| 19 | 0x1<br>RW | **Enable PCIe Link-Down Reset (EN_PLD_RST):** Enable a reset due to a PCIe link-down condition. The PCIe link down condition will cause a HC reset liked. If this bit is set 1, the PCIe link down condition will only reset the PCIe core.<br><br>**Power Well:** SUS |
| 18 | 0x1<br>RW | **Enable EEPROM Reload On Power Up (EN_EEP_REL_PU):** When set to '1' enable EEPROM reload on every main power-up<br><br>**Power Well:** SUS |
| 17 | 0x1<br>RW | **Ignore HC Reset PCIe PHY PIPE (IGN_HC_RST_PPP):** When set to '1' ignore HC reset to the PCIe PHY PIPE reset<br><br>**Power Well:** SUS |
| 16 | 0x1<br>RW | **Ignore LTSSM Reset USB PHY PIPE (IGN_LRST_UPP):** When set to '1' ignore the LTSSM of USB link state transition caused reset to USB PHY PIPE reset<br><br>**Power Well:** SUS |
| 15 | 0x1<br>RW | **Ignore Warm Reset USB PHY Power (IGN_WR_UPP):** When set to '1' ignore warm reset the USB PHY power on reset<br><br>**Power Well:** SUS |
| 14 | 0x1<br>RW | **Allow Core PCIe Link Down Reset (ALL_CPLD_RST):** When set to '1' allow PCIe link down to cause a reset to the rest of the core<br><br>**Power Well:** SUS |
| 13 | 0x0<br>RW | **Ignore Hot Reset USB3 (IGN_HR_U3):** When set to '1' ignore hot reset to the USB3 port logic<br><br>**Power Well:** SUS |
| 12 | 0x0<br>RW | **Ignore Warm Reset USB3 (IGN_WR_U3):** When set to '1' ignore warm reset to the USB3 port logic<br><br>**Power Well:** SUS |
| 11 | 0x0<br>RW | **Ignore Main Power Up Reset USB3 (IGN_MPU_RST_U3):** When set to '1' ignore main power up reset to USB3 port logic<br><br>**Power Well:** SUS |
| 10 | 0x0<br>RW | **Ignore Main Power Up Reset USB2 (IGN_MPU_RST_U2):** When set to '1' ignore main power up reset to USB2 port logic<br><br>**Power Well:** SUS |
| 9 | 0x0<br>RW | **Ignore Main Power Up Reset PCIe Core (IGN_MPU_RST_PC):** When set to '1' ignore main power up reset to PCIe core<br><br>**Power Well:** SUS |
| 8 | 0x0<br>RW | **Ignore Main Power Up Reset PCIe PHY (IGN_MPU_RST_PP):** When set to '1' ignore main power up reset to PCIe PHY<br><br>**Power Well:** SUS |
| 7 | 0x1<br>RW | **Ignore HC Reset USB PHY (IGN_HC_RST_UP):** When set to '1' ignore HC reset to the USB PHY<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6 | 0x1<br>RW | **Ignore Warm Reset USB PHY (IGN_WRST_UP):** When set to '1' ignore warm reset to the USB PHY<br><br>**Power Well:** SUS |
| 5 | 0x1<br>RW | **Enable HC Reset Per Port Isolation (EN_HC_RST_PPI):** Enables the HC reset or per port reset isolation function<br><br>**Power Well:** Core |
| 4 | 0x1<br>RW | **Allow Power Off Power Domain Reset (ALL_PO_PDRST):** When set to '1' allow main power off condition to trigger a main power domain reset<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Ignore Wait For PERST# During Power Show Down (IGN_PERST_PSD):** When set to '1' ignore waiting for PERST# deassertion during main power show down.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Ignore Fundamental Reset During AUX Power Up (IGN_FRST_AUX_PU):** When fundamental reset is asserted during AUX power up, if this bit is set, then we will ignore PERST# such that purely wait for timeout to deassert fundamental reset.<br><br>**Power Well:** SUS |
| 1:0 | 0x0<br>RW | **Trigger Fundamental Reset (TRIG_FRST):** Writing to bit(1:0) to value of 2'b11 will cause a fundamental reset<br><br>**Power Well:** SUS |

## 51.3.147 Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 004A4008h

**BAR and Offset:** [MBAR] + 80C4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:12 | 0x4a4<br>RW | **Max. TT BW Allowed (MAX_TT_BWA):**<br><br>**Power Well:** Core |
| 11:0 | 0x8<br>RW | **Per Packet Overhead SS BW (PP_OVRH_SSBW):** BW calculation: Overhead per packet for SS BW calculations.<br><br>**Power Well:** Core |

## 51.3.148 High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0001A01Fh

**BAR and Offset:** [MBAR] + 80C8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:12 | 0x1a<br>RW | **Per Packet Overhead HS-TT BW (PP_OVRH_HSTTBW):** BW calculation: Overhead per packet for HS-TT BW calculations.<br><br>**Power Well:** Core |
| 11:0 | 0x1f<br>RW | **Per Packet Overhead HS BW (PP_OVRH_HSBW):** BW calculation: Overhead per packet for HS BW calculations.<br><br>**Power Well:** Core |

## 51.3.149 Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00014080h

**BAR and Offset:** [MBAR] + 80CCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:12 | 0x14<br>RW | **Per Packet Overhead FS BW (PP_OVRH_FSBW):** BW calculation: Overhead per packet for FS BW calculations.<br><br>**Power Well:** Core |
| 11:0 | 0x80<br>RW | **Per Packet Overhead LS BW (PP_OVRH_LSBW):** BW calculation: Overhead per packet for LS BW calculations.<br><br>**Power Well:** Core |

## 51.3.150 System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00032010h

**BAR and Offset:** [MBAR] + 80D0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:12 | 0x32<br>RW | **Per TT Packet Overhead System BW (PTTP_OVRH_SBW):** BW calculation: Overhead per TT packet for System BW calculations.<br><br>**Power Well:** Core |
| 11:0 | 0x10<br>RW | **Per Packet Overhead System BW (PP_OVRH_SBW):** BW calculation: Overhead per packet for System BW calculations.<br><br>**Power Well:** Core |

## 51.3.151 Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h

Global defaults for inserting delays between packets in the scheduler for async. types.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 80D4h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 19 | 0x0<br>RW | **High-Speed Bulk Delay Enable (HS_BD_EN):** Reserved.<br><br>**Power Well:** Core |
| 18:16 | 0x0<br>RW | **High-Speed Bulk Delay Default (HS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Full-Speed Bulk Delay Enable (FS_BD_EN):** Reserved.<br><br>**Power Well:** Core |
| 14:12 | 0x0<br>RW | **Full-Speed Bulk Delay Default (FS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **High-Speed Control Delay Enable (HS_CD_EN):** Reserved.<br><br>**Power Well:** Core |
| 10:8 | 0x0<br>RW | **High-Speed Control Delay Default (HS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **Full-Speed Control Delay Enable (FS_CD_EN):** Reserved.<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RW | **Full-Speed Control Default (FS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Low-Speed Control Delay Enable (LS_CD_EN):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Low-Speed Control Delay Default (LS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)<br><br>**Power Well:** Core |

## 51.3.152 AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 808DBCA0h

**BAR and Offset:** [MBAR] + 80E0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **D3 Hot function enable register (D3_HOT_FXN_EN):** This bit is from pin input which is set 1. But we allow software to alter it if it is needed. 1: D3 hot enabled 0: D3 hot not abled.<br><br>**Power Well:** SUS |
| 30 | 0x0<br>RW | **Allow L1 Core Clock Gating (ALL_L1_CORE_CG):** When set to 1 allows core clock being gated during L1 state.<br><br>**Power Well:** SUS |
| 29 | 0x0<br>RW | **Allow Engine PHY Status Extension (AL_EP_SEXT):** When set to 1 allows the engine to extend PHY status of PCIe PIPE for one more cycle. This is due to the fact that our rate change function has a potential of not being able to sample the phystatus signal.<br><br>**Power Well:** SUS |
| 28 | 0x0<br>RW | **Allow Engine PCIe Rate Change Passing (ALL_EP_RCP):** When set to 1 allows the engine to pass PCIe rate change signal as it is from PCIe core to PCIe PHY.<br><br>**Power Well:** SUS |
| 27 | 0x0<br>RW | **Allow Engine PERST Fundamental Reset (AL_PERST_FRST):** When set to 1 allow engine to treat PERST# as a foundamental reset<br><br>**Power Well:** SUS |
| 26 | 0x0<br>RW | **Overwrite PCIe P2 to P1 (OVR_PCIE_P2_P1):** When set to 1 will overwrite a PCIe powerdown state of P2 to P1.<br><br>**Power Well:** SUS |
| 25 | 0x0<br>RW | **Set Internal SSV 1 (SET_ISSV_1):** When set to 1 set the internal SSV to 1.<br><br>**Power Well:** SUS |
| 24 | 0x0<br>RW | **Clear Internal SSV 0 (CLR_ISSV_0):** When set to 1 clear the internal SSV to 0.<br><br>**Power Well:** SUS |
| 23 | 0x1<br>RW | **Enable save_restore_enable SW Loading (EN_SRE_SW_LD):** This is a bit that enables the save_restore_enable signal being loaded when a software command has set Save bit. This is a debug function.<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW | **RESERVED (RSVD_1):** Reserved.<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW | **Force save_restore 1 (FORCE_SR1):** When set to 1, it will force the save_restore flag to 1. This flag is an bit to ensure that we have masked the update during low power state. If software write this bit to 1, it must write it to 0 in order to resume the normal save and restore function.<br><br>**Power Well:** SUS |
| 20 | 0x0<br>RW | **cfg pcie txreg rd (CPTR):** Reserved.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19 | 0x1<br>RW | **cfg iob drivestrength[1] (CIDS1):** Reserved.<br><br>**Power Well:** SUS |
| 18 | 0x1<br>RW | **cfg iob drivestrength[0] (CIDS0):** Reserved.<br><br>**Power Well:** SUS |
| 17 | 0x0<br>RW | **Enable CFG USB P2 (EN_CFG_UP2):** When set to '1' enable cfg usb p2<br><br>**Power Well:** SUS |
| 16 | 0x1<br>RW | **cfg clk gate dis (CCGD):** Reserved.<br><br>**Power Well:** SUS |
| 15 | 0x1<br>RW | **Enable CFG RXDET P3 (EN_CFG_RDP3):** When set to '1' enable cfg rxdet p3<br><br>**Power Well:** SUS |
| 14 | 0x0<br>RW | **Enable CFG PIPE Reset (EN_CFG_PIPE_RST):** When set to '1' enable cfg pipe rst<br><br>**Power Well:** SUS |
| 13 | 0x1<br>RW | **Enable Filter TX Idle (EN_FILT_TX_IDLE):** When set to 1 enables a filter function to TX electrical idle signal at PCie PIPE. We have a filter that will set TXelecidle signal of PCIe PIPE to 1 whenever we are in isolation state or power down transition states.<br><br>**Power Well:** SUS |
| 12 | 0x1<br>RW | **Enable Host Engine Generate PME (EN_HE_GEN_PME):** This is a global switch to whether or not eable this host engine to generate PME message.<br><br>**Power Well:** SUS |
| 11 | 0x1<br>RW | **Enable Isolation (EN_ISOL):** When set to '1' enable isolation<br><br>**Power Well:** SUS |
| 10 | 0x1<br>RW | **Enable L1 Caused P2 Overwrite (EN_L1_P2_OVR):** Set 1 to enable a new feature. This new feature is designed to use L1 as a state to identify whether we should do P2 Overwrite or not. We used to use P1 state to identify whether or not to invoke P2 overwrite function.<br><br>**Power Well:** SUS |
| 9 | 0x0<br>RW | **Enable Core Clock Gating (EN_CORE_CG):** When set to '1' enable core clock gating based on low power state entered<br><br>**Power Well:** SUS |
| 8 | 0x0<br>RW | **Enable PHY Status Timeout (EN_PHY_STS_TO):** When set to '1' enable PHY status timeout function which is designed to cover the PCIePHY issue that we may have not able to detect the PHY status toggle.<br><br>**Power Well:** SUS |
| 7 | 0x1<br>RW | **Ignore aux_pm_en PCIe Core (IGN_APE_PC):** When set to '1' ignore the aux_pm_en reg from PCIe core to continue the remote wake/clock switching support<br><br>**Power Well:** SUS |
| 6 | 0x0<br>RW | **Enable P2 Overwrite P1 (EN_P2_OVR_P1):** When set to '1' enable P2 overwrite P1 when PCIe core has indicated the transition from P0 to P1. This is to enable entering the even lower power state.<br><br>**Power Well:** SUS |
| 5 | 0x1<br>RW | **Enable P2 Remote Wake (EN_P2_REM_WAKE):** When set 1 '1' enable the remote wake function by allowing P2 clock/switching and P2 entering<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4:1 | 0x0<br>RW | **Forced PM State (FORCED_PM_STATE):** Reserved.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Initiate Force PM State (INIT_FPMS):** When set to '1' force PM state to go to the state indicated in bit 4:1<br><br>**Power Well:** SUS |

## 51.3.153 Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 80E4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Enable DM_SRC Battery Charge (EN_DS_BC):** 1 - Always enable battery charge DM_SRC if not connected. Don't wait for portable device detect. (spec. ver. 1.2) 0 - Battery charge spec ver. 1.1.<br><br>**Power Well:** SUS |
| 30:4 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Enable Port 3 Battery Charging (EN_P3_BC):** 0 - Battery charging disabled (Physical Port #3) 1 - Battery charging enabled (Physical Port #3)<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Enable Port 2 Battery Charging (EN_P2_BC):** 0 - Battery charging disabled (Physical Port #2) 1 - Battery charging enabled (Physical Port #2)<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **Enable Port 1 Battery Charging (EN_P1_BC):** 0 - Battery charging disabled (Physical Port #1) 1 - Battery charging enabled (Physical Port #1)<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Enable Port 0 Battery Charging (EN_P0_BC):** 0 - Battery charging disabled (Physical Port #0) 1 - Battery charging enabled (Physical Port #0)<br><br>**Power Well:** SUS |

## 51.3.154 SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 18010000h

**BAR and Offset:** [MBAR] + 80ECh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x3<br>RW | **Force LTSSM State (FORCE_LTSSM_ST):** LTSSM state to be forced This value is for test purpose only.<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW | **Direct Link LTSSM State (DL_LTSSM_ST):** 0: Normal operation mode 1: Direct link to a specific state specified by bit 31:27 This bit is for test purpose only. It shall be written 0 in normal operation mode.<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **Direct Link To U0 (DL_U0):** 0: Normal operation mode 1: Direct link to U0 This bit is for test purpose only. It shall be written 0 in normal operation mode.<br><br>**Power Well:** Core |
| 24:21 | 0x0<br>RW | **Forced Compliance Pattern (FORCED_CMP_PAT):** Compliance pattern to be forced to enter compliance mode This value is for test purpose only.<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW | **Enable Link Error Slave Count (EN_LES_CNT):** 0: Disable link error slave count 1: Enable link error slave count<br><br>**Power Well:** Core |
| 19:17 | 0x0<br>RW | **Debug Mode Select (DEBUG_MD_SEL):** Reserved.<br><br>**Power Well:** Core |
| 16:15 | 0x2<br>RW | **PHY Low Power Latency (PHY_LP_LAT):** This field defines the latency to drive the PHY to enter low power mode 0: 4 cycles 1: 8 cycles 2: 16 cycles 3: 32 cycles<br><br>**Power Well:** Core |
| 14:12 | 0x0<br>RW | **Link Recovery Minimum Time (LR_MIN_TM):** This value defines the minimum time for the link to stay in Recovery.Active other than from U3. The granuity is 128us.<br><br>**Power Well:** Core |
| 11:9 | 0x0<br>RW | **Link Polling Minimum Time (LP_MIN_TM):** This value defines the minimum time for the link to stay in Polling.Active and Recovery.Active from U3. The granuity is 128us.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Force Link Accept PM Command (FORCE_LA_PMC):** 0: Normal operation mode 1: Force link to accept power management command<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **Direct Link Recovery U0 (DL_REC_U0):** 0: Normal operation mode 1: Direct link to Recovery from U0<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Link Fast Training Mode (LINK_FTM):** 0: Normal operation mode 1: Link fast training mode This bit should be written 0 in normal operation.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **Disable Link Scrambler (DIS_LINK_SCRAM):** 0: Enable link scrambler 1: Disable link scramber<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW | **Direct Link U3 From U0 (DL_U3_U0):** 0: Normal operation mode 1: Direct link to U3 from U0 This bit is for test purpose only. It shall be written 0 in normal operation mode.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Direct Link U3 From U0 (DL_U2_U0):** 0: Normal operation mode 1: Direct link to U2 from U0 This bit is for test purpose only. It shall be written 0 in normal operation mode.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **Direct Link U3 From U0 (DL_U1_U0):** 0: Normal operation mode 1: Direct link to U1 from U0 This bit is for test purpose only. It shall be written 0 in normal operation mode.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Enable Link Loopback Master Mode (EN_LINK_LB_MAST):** 0: Disable link loopback master mode 1: Enable link loopback master mode<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Disable Link Compliance Mode (DIS_LINK_CM):** 0: Enable link compliance mode 1: Disable link compliance mode<br><br>**Power Well:** Core |

## 51.3.155 USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 80F0h

**Default:** 310803A0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x31<br>RW | **FS/LS Mode SE0 Disconnect Delay[7:0] (FSLS_SE0_DIS_DEL_7_0):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.<br><br>**Power Well:** SUS |
| 23:21 | 0x0<br>RW | **Reserved (RSVD0):** Reserved.<br><br>**Power Well:** SUS |
| 20 | 0x0<br>RW | **L1_EXIT_RECOVERY_MODE:** Mode for extended L1 Exit recovery delay: 0: 12us 1: 50us<br><br>**Power Well:** SUS |
| 19 | 0x1<br>RW | **L1_TO_INCR_MODE:** Mode select for L1 Timeout increments: 0: time out increments are in 125us 1: L1 Timeout increments are in 256us. Refer to USB2 PORTHLPMC.L1 Timeout in XHCI Spec for additional details<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW | **Reserved (RSVD1):** Reserved.<br><br>**Power Well:** SUS |
| 17 | 0x0<br>RW | **EN_DETECT_NOMINAL_PKT_EOP:** 0: Detect minimal packet EOP. 1: Detect nominal packet EOP.<br><br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **Disable Chirp Response (DIS_CHIRP_RESPONSE):** 0: Normal 1: Force full speed on host ports (disable chirp response)<br><br>**Power Well:** SUS |
| 15 | 0x0<br>RW | **Disable 192 Byte Limit Check (DIS_192B_LIM):** 0: Enforce 192 byte limit on complete-split INs. Treat any packet ) 192 as babble case. 1: Disable 192 byte limit check.<br><br>**Power Well:** SUS |
| 14 | 0x0<br>RW | **External Provided FS/LS Disconnect (EXT_FSLS_DIS):** 0: Internal FS/LS Disconnect from linestate(1:0) 1: External provided FS/LS Disconnect from hostdisconnect input<br><br>**Power Well:** SUS |
| 13:12 | 0x0<br>RW | **UTMI Reset Source Select (UTMI_RST_SEL):** Select UTMI Reset Source (FRD UTMI Reset Only) 00: HCReset or Force PHY Reset or internal reset after disconnect/suspend for restart (default) 01,11: UTMI reset = ~UTMI suspendm 10: UTMI reset = ~UTMI suspendm and synchronization to port clk.<br><br>**Power Well:** SUS |
| 11 | 0x0<br>RW | **Disable HS Disconnect Window (DIS_HS_DIS_WIN):** 0: Enable HS Disconnect Window Function 1: Disable HS Disconnect Window Function<br><br>**Power Well:** SUS |
| 10 | 0x0<br>RW | **Disable Port Error Detection (DIS_PERR_DET):** 0: Enable Port Error Detection (default) 1: Disable Port Error Detection<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 9 | 0x1<br>RW | **Disable Peek Function for ISO-OUT (DIS_PF_IOUT):** 0: Enable Peek function for ISO-OUT (default) 1: Disable Peek function for ISO-OUT<br><br>**Power Well:** SUS |
| 8 | 0x1<br>RW | **Drive Resume-K FS/LS Serial Interface (DRV_RESK_FSLS_SER):** 0: Drive Resume-K on parallel Interface 1: Drive Resume-K directly on FS/LS Serial Interface (default)<br><br>**Power Well:** SUS |
| 7 | 0x1<br>RW | **Enable USB2 Drop-Ping (EN_U2_DROP_PING):** 0: Disable Drop-Ping Function in USB2 Protocol (default) 1: Enable Drop-Ping Function in USB2 Protocol<br><br>**Power Well:** SUS |
| 6 | 0x0<br>RW | **Enable USB2 Force-Ping (EN_U2_FORCE_PING):** 0: Disable Force-Ping Function in USB2 Protocol (default) 1: Enable Force-Ping Function in USB2 Protocol<br><br>**Power Well:** SUS |
| 5 | 0x1<br>RW | **Enable USB2 Auto-Ping (EN_U2_AUTO_PING):** 0: Disable Auto-Ping Function 1: Enable Auto-Ping Function in USB2 Protocol (default)<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW | **Disable PHY SuspendM (DIS_PHY_SUSM):** 0: PHY is suspend=U3,U2,disconnect (default) 1: Disable PHY SuspendM in All States<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **UTMI Internal Clock Gate Disable (UTMI_INT_CG_DIS):** 0: Normal operation (internal clock gated in U2,U3,disconnect) 1: UTMI Internal Clock Gate Disable<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Disable PHY SuspendM in Disconnect State (DIS_PSUSM_DS):** 0: PHY is suspendM=0 in Disconnect State (default) 1: Disable PHY SuspendM in Disconnect State<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **Force PHY Reset (FORCE_PHY_RST):** 0: Normal Operation (default) 1: Force PHY Reset<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **USB2 Accelerated Simulation Timing (U2_ACC_SIM_TIM):** 0: Normal Operation (default - FPGA/ASIC) 1: USB2 Accelerated Simulation Timing (default - simulation)<br><br>**Power Well:** SUS |

## 51.3.156 USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 80F4h

**Default:** 80C40620h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **Total Reset Duration[0] (TOT_RST_DUR_0):** # of microseconds for total reset duration<br><br>**Power Well:** SUS |
| 30:18 | 0x31<br>RW | **Chirp-K Duration (CHIRPK_DUR):** # of microseconds of Chirp-K to register that a device is chirping<br><br>**Power Well:** SUS |
| 17:5 | 0x31<br>RW | **K/J Disconnect Connect Delay (KJ_DIS_CON_DEL):** # of microseconds of K/J in disconnected state to register connect has occurred.<br><br>**Power Well:** SUS |
| 4:0 | 0x0<br>RW | **FS/LS Mode SE0 Disconnect Delay[12:8] (FSLS_SE0_DIS_DEL_12_8):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.<br><br>**Power Well:** SUS |

## 51.3.157 USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 80F8h

**Default:** F865EB6Bh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf<br>RW | **U2 Entry Ignore Linestate Changes Duration[3:0] (U2_IGN_LS_DUR_3_0):** # of microseconds after entering U2, linestate changes are ignored as bus settles<br><br>**Power Well:** SUS |
| 27:15 | 0x10cb<br>RW | **U3 Entry Ignore Linestate Changes Duration (U3_IGN_LS_DUR):** # of microseconds after entering U3, linestate changes are ignored as bus settles<br><br>**Power Well:** SUS |
| 14:0 | 0x6b6b<br>RW | **Total Reset Duration[15:1] (TOT_RST_DUR_15_1):** # of microseconds for total reset duration<br><br>**Power Well:** SUS |

## 51.3.158 USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00008003h

**BAR and Offset:** [MBAR] + 80FCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 21:9 | 0x40<br>RW | **U2 Detect Remote Wake Delay (U2D_RWAKE_DEL):** #of microseconds after detecting U2 remote wake condition to reflect K<br><br>**Power Well:** SUS |
| 8:0 | 0x3<br>RW | **U2 Entry Ignore Linestate Changes Duration[12:4] (U2_IGN_LS_DUR_12_4):** # of microseconds after entering U2, linestate changes are ignored as bus settles<br><br>**Power Well:** SUS |

## 51.3.159 Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00008020h

**BAR and Offset:** [MBAR] + 810Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:12 | 0x8<br>RW | **Per Burst Overhead System BW (PB_OVRH_SBW):** BW calculation: Overhead per burst for system BW calculations.<br><br>**Power Well:** Core |
| 11:0 | 0x20<br>RW | **Per Burst Overhead System BW (PB_OVRH_SSBW):** BW calculation: Overhead per burst for SS BW calculations.<br><br>**Power Well:** Core |

## 51.3.160 USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0F42528505647F42h

**BAR and Offset:** [MBAR] + 8128h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:60 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 59:48 | 0xf42<br>RW | **PCIe Max BW Units (PCIE_MAX_BW):** Max. Number of BW units for PCIe (system interface) (denominator in 90% calculation)<br><br>**Power Well:** Core |
| 47:36 | 0x528<br>RW | **TT Max BW Units (TT_MAX_BW):** Max. Number of BW units for TTs. (denominator in 90% calculation)<br><br>**Power Well:** Core |
| 35:24 | 0x505<br>RW | **FS/LS Max BW Units (FSLS_MAX_BW):** Max. Number of BW units for FS/LS ports. (denominator in 90% calculation)<br><br>**Power Well:** Core |
| 23:12 | 0x647<br>RW | **HS Max BW Units (HS_MAX_BW):** Max. Number of BW units for HS ports. (denominator in 80% calculation)<br><br>**Power Well:** Core |
| 11:0 | 0xf42<br>RW | **SS Max BW Units (SS_MAX_BW):** Max. Number of BW units for SS ports. (denominator in 90% calculation)<br><br>**Power Well:** Core |

## 51.3.161 Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0A019132h

**BAR and Offset:** [MBAR] + 8140h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0xa<br>RW | **Engine Idle Hysteresis (EIH):** This register controls the min. idle span that has to be observed from the engine idle indicators before the power state flags (xhc_*_idle) will indicate a 1.<br><br>**Power Well:** Core |
| 23:12 | 0x19<br>RW | **Backbone PLL Shutdown Advance Wake (BPSAW):** This register controls the time before the next scheduled transaction where the Backbone PLL request will assert. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11:0 | 0x132<br>RW | **Backbone PLL Shutdown Min. Idle Duration (BPSMID):** The sum of this register plus the Backbone PLL Shutdown Advance Wake form to a Total Idle time. When the next scheduled periodic transaction is after present time + Total Idle, the Backbone PLL request will de-assert, allowing the PLL to shutdwon. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)<br><br>**Power Well:** Core |

## 51.3.162 AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 81390206h

**BAR and Offset:** [MBAR] + 8154h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **DIS_L1P2_EXIT_ON_WAKE_EN:** This bit disables the dependency on Wake Enables defined in PORTSC for L1P2 exit when in D0<br><br>**Power Well:** SUS |
| 30:28 | 0x0<br>RW | **RESERVED0 (RSVD0):** Reserved.<br><br>**Power Well:** SUS |
| 27:25 | 0x0<br>RW | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 24 | 0x1<br>RW | **Enable L1 exit notification to SNPS PCIe core (EN_L1_EXIT_NOTIF_PCIE):** This bit enables a L1 exit notification to SNPS PCIe core. There is a case where USB ports have waked up and AUX PM module has started the wakeup process. The AUX PM control state got into a wait for P0 state because it needs to wait until PCie core to signal powerdown state change. Due to the fact that the core is in D3Hot, there is no run_stop bit set such that no internal interrupt will be fired. This causes the LTSSM of PCIe stayed in L1 even though AUX PM has known that it needs an L1 exit. This bit works together with bit21 of this register. 1: enables this feature 0: disables this feature.<br><br>**Power Well:** SUS |
| 23 | 0x0<br>RW | **DISABLE PLC ON DISCONNECT (DIS_PLC_ON_DISCONNECT):** 1: do not assert PLC for disconnection 0: assert PLC for disconnection<br><br>**Power Well:** SUS |
| 22 | 0x0<br>RW | **TREAT_IDLE_AS_TS2_IN_LTSSM_WAIT_4_TS2:** This bit enables a feature in PCie core LTSSM to treat IDLE received as TS2 when LTSSM is in wait for TS2 receive state. This is a function requested from PHY where it is possible to not able to receive TS2 without error. 1: treat Logic IDLE as TS2 received when in some PCIe LTSSM state. 0: disable this feature.<br><br>**Power Well:** SUS |
| 21 | 0x1<br>RW | **Disable p2 overwrite due to the D3HOT where PCIe core enters the L1 (DIS_P2_OVERWRITE_DUE2_D3HOT):** We added a feature where if PCIe core LTSSM enters L1 due to the D3hot, the aux PM control will not start a P2 overwrite function in anticipating for the next L23 enter. 1: disables p2 overwrite due to the D3HOT where PCIe core enters the L1. 0: enables P2 overwrite even if we are in D3Hot.<br><br>**Power Well:** SUS |
| 20 | 0x1<br>RW | **Enable the port to enter U3 automatically when in U1/U2 (ENABLE_AUTO_U3_ENTRY_FROM_U2_U3):** 1: enables the port to enter U3 automatically when in U1/U2 0: disables the port to enter U3 automatically when in U1/U2<br><br>**Power Well:** SUS |
| 19 | 0x1<br>RW | **No linkdown reset is issue during low power state (DIS_LINKDOWN_RST_DURING_LOW_POWER):** No linkdown reset is issue during low power state<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW | **EN_EXIT_DEEP_SLEEP_IF_PCIE_IN_P0:** This bit enables a feature in AUX PM module where if PCIe core LTSSM is in P0 for a duration of time, we will exit the deep sleep state. This is for failure control in case. 1: enables this feature 0: disable this feature<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **U2_EXIT_LFPS_TIMER_VALUE:** This bit selects U2 exit LFPS timer value 0: 320ns 400ns in 25MHz domain 1: 240ns 320ns in 25MHz domain<br><br>**Power Well:** SUS |
| 16 | 0x1<br>RW | **EN_EXIT_DEEP_SLEEP_ON_USB_PORT_WAKEUP:** This bit enables a function that AUX PM module exits from the deep sleep state due to the USB ports wakeup level signal. We have added this feature where USB ports will generated a wakeup level signal to wakeup the AUX PM module if it is in deep sleep state and this level signal will be cleared if the change bits are updated by software. 1: enables this function 0: disables this function which means that a wakeup pulse generated from each USB PortSC event will wake up the AUX PM module from deep sleep if the D3 state is programmed.<br><br>**Power Well:** SUS |
| 15:14 | 0x0<br>RW | **P3_ENTRY_TIMEOUT:** This field defines the timeout value to enter P3 mode in U2. 00: 7us 8us 01: 511us 512us 10: disables the timer (0us) 11: disables the timer (0us)<br><br>**Power Well:** SUS |
| 13 | 0x0<br>RW | **Enable U2 P3 Mode (EN_U2_P3):** 0: Disable U2 P3 mode 1: Enable U2 P3 mode<br><br>**Power Well:** SUS |
| 12:11 | 0x0<br>RW | **Fine Debug Mode Select (FINE_DM_SEL):** Reserved.<br><br>**Power Well:** SUS |
| 10 | 0x0<br>RW | **Enable Low Power State Based Core Clock Gating (EN_LP_CORE_CG):** When set to '1' enable core clock gating based on low power state entered<br><br>**Power Well:** SUS |
| 9 | 0x1<br>RW | **Disable USB3 Port Status Changed Event (DIS_U3_PORT_SCE):** 0: Enable USB3 port status change event generation if any change bit is not cleared 1: Disable USB3 port status change event generation if any change bit is not cleared Bit 12 default 0<br><br>**Power Well:** SUS |
| 8:4 | 0x0<br>RW | **Debug Mode Select Register (DEB_MODE_SEL):** Reserved.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Enable Auto Wakeup Non-IDLE (EN_AWAK_NIDLE):** When set to 1 enables the auto wakeup function when engine has identified non IDLE condition.<br><br>**Power Well:** SUS |
| 2 | 0x1<br>RW | **Enable PM Control P1 Exit P2 (EN_PMC_P1_EXIT_P2):** When set 1 enables the PM control module to transition to P1 instead of P0 when exit P2.<br><br>**Power Well:** SUS |
| 1 | 0x1<br>RW | **Enable PCIe PIPE CLK Isolation (EN_PP_CLK_ISOL):** When set to 1 enables the PCIe PIPE CLK to be isolated when main power is removed.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **Enable P2 Overwrite P1 Allowed Detect (EN_P2OVRP1_ADET):** When set to 1 enables a function that can detect whether or not enable P2 overwrite P1 function. The condition to get to P2 overwrite is when engine is in idle conditions. This means that there is no ISO EP pending.<br><br>**Power Well:** SUS |

## 51.3.163 USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000000FCh

**BAR and Offset:** [MBAR] + 8164h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 7 | 0x1<br>RW | **EN_CMDM_TXRXB:** Enable Command Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 6 | 0x1<br>RW | **EN_TTE_TXRXB:** Enable TTE Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 5 | 0x1<br>RW | **EN_IDMA_TXRXB:** Enable IDMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 4 | 0x1<br>RW | **EN_ODMA_TXRXB:** Enable ODMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 3 | 0x1<br>RW | **EN_TRM_TXRXB:** Enable Transfer Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 2 | 0x1<br>RW | **EN_SCH_TXRXB:** Enable Scheduler Active indication for Tx/Rx Bias circuit HS Phy PM Policy<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Enable Rx Bias ckt disable (EN_RXB_CD):** When set enables the Rx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Enable Tx Bias ckt disable (EN_TXB_CD):** When set enables the Tx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)<br><br>**Power Well:** Core |

## 51.3.164 USB Power Gating Control (USB_PGC)—Offset 8168h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000282EEh

**BAR and Offset:** [MBAR] + 8168h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x14177<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **USB SRAM power gating enable (USB_SRAM_PGE):** When set enables power gating on USB ports. Usage of this bit is further qualified with xHCI SRAM Dynamic Power Gating Disable fuse. If the fuse disables dynamic power gating, setting this bit to 1 shall not enable power gating feature. This bit always returns the value that was written to it irrespective of the setting of xHCI SRAM Dynamic Power Gating Disable fuse.<br><br>**Power Well:** Core |

## 51.3.165 xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000400h

**BAR and Offset:** [MBAR] + 816Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 19 | 0x0<br>RW | **USB3 Partition Engine/Link trunk gating Enable (PARUSB3_ENG_GEN):** When set to 1 enables gating of the SOSC trunk to the XHCI engine and link in the PARUSB3 partition.<br><br>**Power Well:** SUS |
| 18 | 0x0<br>RW | **USB3 Partition Frame Timer trunk gating Enable (PARUSB3_LINK_GEN):** When set to 1 enables gating of the SOSC trunk to the Frame timer in the PARUSB3 partition.<br><br>**Power Well:** SUS |
| 17 | 0x0<br>RW | **USB2 link partition clock gating enable (PARUSB2_CLK_GEN):** When set to 1 enables gating of the SOSC trunk to the USB2 link and Phy logic in the PARUSB2 partition.<br><br>**Power Well:** SUS |
| 16 | 0x0<br>RW | **USB2/USHIP 12.5 MHz partition clock gating enable (USHIP_PCGEN):** When set to 1 enables gating of the 12.5 MHz SOSC trunk to the USB2 and USHIP logic in the PARUSB2 partition.<br><br>**Power Well:** SUS |
| 15 | 0x0<br>RO | **Reserved1 (RSVD1):** Reserved |
| 14 | 0x0<br>RW | **USB3 Port Aux/Core clock gating enable (USB3_AC_CGE):** When set, allows the aux_cclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 13:12 | 0x0<br>RW | **Rx Detect Timer when port Aux Clock is Gated (RX_DT_ACG):** This field defines the value of the timer used to perform Rx Detect when port Aux Clock has been gated. 0x0: 100ms 0x1: 12ms Others: Reserved Note: This timer shall use the Fast Training Timer Tick (about 1us tick) for simulation purposes. For Fast Training mode, the above timeouts will become about 11us and about 100us, +/- implementation uncertainty, respectively.<br><br>**Power Well:** SUS |
| 11:8 | 0x4<br>RW | **U2 Residency Before ModPHY Clock Gating (U2R_BM_CG):** Before gating ModPHY Aux clock, Host Controller shall wait for this time in U2. This time is meant to ensure that the attached device has entered U2 as well. 0x0: 1us 0x1: 128us 0x2: 256us 0x3: 512us 0x4: 640us 0x5: 768us 0x6: 896us 0x7: 1024us Others: Reserved Note: This counter shall start counting once pipe has entered PS3 state in response to link in U2.<br><br>**Power Well:** SUS |
| 7 | 0x0<br>RW | **Frame Timer Clock Gating Ports in U2 Enable (FTCGPU2E):** This bit, when set, allows Host Controller to gate the clock to the Frame Timer when ports are in U2.<br><br>**Power Well:** SUS |
| 6 | 0x0<br>RW | **USB2 port clock throttle enable (USB2_PC_TE):** When set, allows the Aux clock into the USB2 ports to be throttled when conditions allow.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **XHCI Engine Aux clock gating enable (XHCI_AC_GE):** When set, allows the aux clock into the XHCI engine to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 4 | 0x0<br>RW | **XHCI Aux PM block clock gating enable (XHCI_APMB_CGE):** When set, allows the aux clock into the Aux PM block to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **USB3 Aux Clock Trunk Gating Enable (USB3_AC_TGE):** When set, allows Aux Clock Trunk feeding to USB3.0 ports to be gated when port Aux clock is gated at all USB3.0 ports and all USB3.0 modPHY instances. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **USB3 Port Aux/Port clock gating enable (USB3_AP_CGE):** When set, allows the aux_pclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **ModPHY port Aux clock gating enable in U2 (MPP_AC_GEU2):** When set, allows the aux clock into the ModPhy to be gated when Link is in U2 and pipe has been in PS3 for at least the time defined by U2 Residency Before ModPHY Clock Gating field. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **ModPHY port Aux clock gating enable in Disconnected, U3 or Disabled (MPP_AC_GE_DDU3):** When set, allows the aux clock into the ModPHY to be gated when Link is in Disconnected, U3 or Disabled state. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.<br><br>**Power Well:** SUS |

## 51.3.166 USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 96090032h

**BAR and Offset:** [MBAR] + 8170h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x258<br>RW | **USB2_LPM_REG_RSM_U3_DET_NORM:** Reserved.<br><br>**Power Well:** Core |
| 21:19 | 0x1<br>RW | **Min U3 Exit LFPS Duration (MIN_U3E_LFPS_D):** Min U3 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U3 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us<br><br>**Power Well:** SUS |
| 18:16 | 0x1<br>RW | **Min U2 Exit LFPS Duration (MIN_U2_ELFPS_D):** Min U2 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U2 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us<br><br>**Power Well:** SUS |
| 15 | 0x0<br>RW | **Max PING LFPS Rx Detection (XHCI_MAX_PING_LFPS):** This field defines the maximum timing for PING LFPS. If an incoming LFPS will be considered a PING if it has a timing such that it is less than or equal to the selected value. Otherwise it will be considered for the other types of LFPS. 0b Max PING LFPS timing set to 256 ns (32 link clocks) 1b Max PING LFPS timing set to 320 ns (40 link clocks)<br><br>**Power Well:** SUSe |
| 14:10 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 9:0 | 0x32<br>RW | **xHCI BESL to HIRD Distance (XHCI_BESL_HIRD_DT):** This field defines the gap between BESL and duration of Resume signalling from Host upon Host Initiated Resume from USB2.0 LPM. Default value of this register corresponds to xHCI spec defined 50us value. Value BESL to HIRD Distance 000h 0us 001h 1us 002h 2us 3FFh 1023us<br><br>**Power Well:** SUS |

## 51.3.167 xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0040047Dh

**BAR and Offset:** [MBAR] + 8174h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Disable scheduler direct transition from IDLE to NO requirement (DIS_SDT_IDL_NR):** 0: (default) allow scheduler direct transition from IDLE to NO requirement 1: Disable scheduler direct transition from IDLE to NO requirement<br><br>**Power Well:** Core |
| 30:26 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **XHCI LTR Transition Policy (XLTRTP):** When '0', LTR messaging state machine transitions from High, Medium, or Low LTR states to Active state upon the Alarm Timer timout and stays in Active until the next service boundary. When '1', the LTR messaging state machine transitions through High ? Med ? Low ? Active states assuming enough latency is available for each transition.<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **XHCI LTR Enable (XLTRE):** This bit must be set to enable LTV messaging from XHCI to the PMC.<br><br>**Power Well:** Core |
| 23:12 | 0x400<br>RW | **Periodic Active LTV (PA_LTV):** 23:22 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 21:12 Latency Value (ns) Defaults to 0 micro seconds<br><br>**Power Well:** Core |
| 11:0 | 0x47d<br>RW | **USB2 Port L0 LTV (USB2_PL0_LTV):** 11:10 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 9:0 Latency Value (ns) Defaults to 128 Micro Seconds<br><br>**Power Well:** Core |

## 51.3.168 xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000017FFh

**BAR and Offset:** [MBAR] + 8178h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:13 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x17ff<br>RW | **LTV Limit (LTV_LMT):** This register defines a maximum LTR value that is allowed to be advertised to the PMC. This is meant to be used as a workaround or mitigation if issues are discovered with the LTR values generated by the XHC using the defined algorithms. If the LTR value of the XHC is larger than the value in this register field, the value in this field is sent to the PMC instead. Default value is the highest possible - 101b 12:10: Latency Multiplier Field 000b - Value times 1 ns 001b - Value times 32 ns 010b - Value times 1,024 ns 011b - Value times 32,768 ns 100b - Value times 1,048,576 ns 101b - Value times 33,554,432 ns 110b-111b - Not Permitted 9:0: Latency Value Default = 3FFh<br><br>**Power Well:** Core |

## 51.3.169 xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 817Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 28:16 | 0x0<br>RW | **Minimum High Idle Time (MHIT):** LTR value can be indicated. This value must be larger than HIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |
| 15:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RW | **High Idle Wake Latency (HIWL):** This is the latency to access memory from the High Idle Latency state. This value must be larger than MIWL and LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |

## 51.3.170 xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8180h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 28:16 | 0x0<br>RW | **Minimum Medium Idle Time (MMIT):** LTR value can be indicated. This value must be larger than MIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |
| 15:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RW | **Medium Idle Wake Latency (MIWL):** This is the latency to access memory from the Medium Idle Latency state. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |

## 51.3.171 xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8184h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 28:16 | 0x0<br>RW | **Minimum Low Idle Time (MLIT):** LTR value can be indicated. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |
| 15:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RW | **Low Idle Wake Latency (LIWL):** This is the latency to access memory from the Medium Idle Latency state. 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds ( 0 - 124 Micro Seconds)<br><br>**Power Well:** Core |

## 51.3.172 HOST_CTRL_BW_MAX3_REG—Offset 8188h

Added for CHV

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00F42F42h

**BAR and Offset:** [MBAR] + 8188h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 23:0 | 0xf42f42<br>RW | **MAX_OUT_BW_UNITS_FOR_SS_PORTS:** Max. Number of OUT BW units for SS ports - denominator in 90% calculation<br><br>**Power Well:** Core |

## 51.3.173 THRM_HOST_CTRL_REG—Offset 819Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 819Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RW | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW | **SSIC Thermal Throttle Ux Mapping (SSIC_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.<br><br>**Power Well:** Core |
| 19:18 | 0x0<br>RW | **USB3 Thermal Throttle Ux LGO delay (USB3_THRM_UX_LGO_DELAY):** Controls the delay enforced between LMP for FLPMA and LMP for Ux LGO. After sending LPMA ON , wait for pre-defined number of clocks to initiate LGO_U1/LGO_U2 00: 8 clocks 01: 32 clocks 10: 128 clocks 11: 0 clocks This field does not apply to ports that are not operating in the mode required to issue FLMA ON.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW | **THRM_THROTTLING_DISABLE:** 0: Thermal throttling is enabled. 1: Thermal throtling is disabled. The host controller ignores the TT control inputs and does not throttle.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **USB3 Thermal Throttle Ux Mapping (USB3_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **THROTTLE_PRIORITY_MODE:** 0: Off period has priority: In this case, when the throttle signal is asserted, the host controller enters the off state on the next uframe boundary, and stays in the off state for the prescribed duration or until the end of the 16 uframe throttle period whichever occurs first. On subsequent throttle periods, the off period occurs first and then the on period. 1: The On period has priority: In this case, when the throttle signal is asserted, the host controller completes the required On period first before entering the off period. If the required number of uFrames has already been executed in a 16 uframe throttle window, the controller enters the off period immediately.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW | **DISABLE_FORCE_L1_WHEN_THROTTLED:** 0: USB2 port will not force L1 entry on throttled ports. L1 entry will be based on the normal idle timeout 1: USB2 ports will attempt to enter L1 immediately after throttled ports are idle.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **DISABLE_INTERRUPT_THROTTLING:** 0: Interrupt traffic is throttled 1: Interrupt traffic is not throttled<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW | **DISABLE_ISOCHRONOUS_THROTTLING:** 0: Isochronous traffic is throttled 1: Isochronous traffic is not throttled<br><br>**Power Well:** Core |
| 11:8 | 0x0<br>RW | **T1_ACTION:** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW | **T2_ACTION:** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **T3_ACTION:** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.<br><br>**Power Well:** Core |

## 51.3.174 THRM_HOST_CTRL_REG2—Offset 81B4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0000000Fh

**BAR and Offset:** [MBAR] + 81B4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 3:0 | 0xf<br>RW | **Force LPM Accept Enable (FORCE_LPM_ACCEPT_EN):** Per Port Control to allow for enforcement to be based on device detection if certain devices do not support FLPMA. 0: Do not set FLPMA prior to Ux entry due to TT 1: Set FLPMA prior to Ux entry due to TT<br><br>**Power Well:** Core |

## 51.3.175 LFPSONCOUNT_REG—Offset 81B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000020C8h

**BAR and Offset:** [MBAR] + 81B8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** SUS |
| 17:16 | 0x0<br>RW | **U2P3 LFPS Periodic Sampling Control (XU2P3LPSC):** This field controls the OFF time for the LFPS periodic sampling for SS and SSIC ports in U2P3. If LFPSPM for a port is 1, it will override the OFF time and LFPS receiver will remain OFF permanently. For Fast Sim mode, 500us will be equivalent to 5us. 0x0 Polling Disable. (RXDET Polling will become 100ms.) 0x1 500us OFF Time 0x2 1ms OFF Time 0x3 1.5ms OFF Time<br><br>**Power Well:** SUS |
| 15:10 | 0x8<br>RW | **XLFPSONCNTSSIC:** This time would describe the number of clocks SSIC LFPS will remain ON. SSIC LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 8.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 9:0 | 0xc8<br>RW | **XLFPSONCNTSS:** This time would describe the number of clocks LFPS will remain ON. LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 200.<br><br>**Power Well:** SUS |

## 51.3.176 reg_D0i2_CTRL_REG_type (D0i2_CTRL_REG)—Offset 81BCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 084204B0h

**BAR and Offset:** [MBAR] + 81BCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 29:26 | 0x2<br>RW | **D0i2 Minimum Residency (D0I2_MIN_RESIDENCY):** This field controls the minimum time that we must stay in D0i2 to ensure that the entry sequence has settled before we attempt to exit. 0h Disabled 1h 8 clocks 2h 16 clocks 3h 32 clocks 4h 128 clocks 5h 256 clocks 6h 512 clocks 7h 1024 clocks<br><br>**Power Well:** SUS |
| 25:22 | 0x1<br>RW | **D0i2 Entry Hysteresis Timer (D0I2_ENTRY_HYSTERESIS_TIMER):** This field allows for a hysteresis timer to be implemented specifically for D0i2. This will allow for D0i2 entry to be controlled independently from the timer used for D0i3 and D3. 0h Disabled 1h 8 clocks 2h 16 clocks 3h 32 clocks 4h 64 clocks 5h 128 clocks 6h 256 clocks 7h 512 clocks<br><br>**Power Well:** SUS |
| 21 | 0x0<br>RW | **Active Periodic EP Disable (ACTIVE_PERIODIC_EP_DISABLE):** This field allows the xHC to control how aggressive it enters D0i2 in the presence of active Periodic EPs. Setting this field will allow D0i2 only when there are no active Periodic EPs on the schedule. Either there are no active DB or any active Interrupt EP is flow controlled.<br><br>**Power Well:** SUS |
| 20:16 | 0x2<br>RW | **MSI D0i2 Pre Wake Time (MSID0I2PWT):** This is the latency that is expected to be incurred to exit the D0i2 state. This wake latency is the latency to be added to the tracked D0i2 wake by the MSI module. Example: If while allowing D0i2 there is an MSI generation that will trigger in 250 us from now ,the MSI module will trigger a D0i2 wake up 250 us MSI D0i2 Pre Wake Time. The D0I2WL is to ensure that the act of D0i2 exit does not impact the action(s) required by the alarm(s) that was enabled prior to D0i2. 3:0 250 ns ticks 0h 0 ns 1h 250 ns . 1Fh 3750 ns<br><br>**Power Well:** SUS |
| 15:4 | 0x4b<br>RW | **MSI Idle Threshold (MSI_IDLE_THRESHOLD):** This field allows the xHC to control how aggressive it enters D0i2 in the presence of pending MSI. This field is valid only if Pending MSI Disable is 0, allowing D0i2 in the presence of pending MSIs. D0i2 will be prevented if the amount of idle time between Event Ring being idle to the time an MSI will be generated does not exceed this time. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Pending MSI Disable (PENDING_MSI_DISABLE):** This field allows the xHC to disable D0i2 when there are pending MSIs in the event manager. Setting this field will require all IMOD counters to be 0h and all MSIs delivered. Clearing this field will allow for D0i2 power gating while there are 1 or more IMOD counters decrementing which implies that an MSI is pending and will be generated once the corresponding IMOD counter reaches 0h.<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Frame Timer Run Disable (FRAME_TIMER_RUN_DISABLE):** This field allows the xHC to disable D0i2 when the frame timer is running. Clearing this field will allow D0i2 when the frame timer is required as defined in the XHCI Spec. Setting this field will not allow D0i2 when the frame timer is required and will limit D0i2 for the condition where the frame timer is not required.<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **USB2 L1 Disable (USB2_L1_DISABLE):** This field allows the xHC to disable D0i2 when USB2 ports are in L1. This implies that D0i2 will only be triggered when ports are in L2 or deeper.<br><br>**Power Well:** SUS |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RW | **USB3 U2 Disable (USB3_U2_DISABLE):** This field allows the xHC to disable D0i2 when USB3 ports are in U2. This implies that D0i2 will only be triggered when ports are in U3 or deeper.<br><br>**Power Well:** SUS |

## 51.3.177 reg_D0i2_SCH_ALARM_CTRL_REG_type (D0i2_SCH_ALARM_CTRL_REG)—Offset 81C0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 81C0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 28:16 | 0x0<br>RW | **D0i2 Idle Time (D0I2IT):** This is the minimum schedule idle time that must be available before D0i2 can allowed. 12:7 Time value in # of 125s Bus Intervals (0 8ms) 6:0 Fractional BI Time value in s ( 0 124s) This field controls how much scheduler idle time is required to trigger D0i2. The scheduler idle time is the same idle time that is used by other functions implementing alarm timers such as PLL shutdown, LTR and USB2 L1 override. If there is a need to require other alarm timers to have been set such as LTR has triggered a Low Idle time, then this field needs to be as aggressive as LTR Low Min Idle Time to ensure D0i2 is triggered once the scheduler has detected IDLE in the schedule. This is only to be used as alternate modes of operation or back up modes.<br><br>**Power Well:** SUS |
| 15:13 | 0x0<br>RO | **Reserved (RSVD1):** Reserved.<br><br>**Power Well:** SUS |
| 12:0 | 0x0<br>RW | **D0i2 Wake Latency (D0I2WL):** This is the latency that is expected to be incurred to exit the D0i2 state. This wake latency is the latency to be added to the tracked D0i2 wake by the scheduler. Example: If while allowing D0i2 there is an alarm that will trigger in 250 us from now , the scheduler will allow D0i2 and track this alarm with an adjustment to wake up 250ns D0I2WL) from. The D0I2WL is to ensure that the act of D0i2 exit does not impact the action(s) required by the alarm(s) that was enabled prior to D0i2. 12:7 Time value in # of 125s Bus Intervals (0 8ms) 6:0 Fractional BI Time value in s (0124s)<br><br>**Power Well:** SUS |

## 51.3.178 reg_USB2PMCTRL_REG_type (USB2PMCTRL_REG)—Offset 81C4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 81C4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** SUS |
| 11 | 0x0<br>RW | **USB2 PHY SUS Power Gate PORTSC Block Policy (U2PSPGPSCBP):** This controls the policy for blocking PORTSC Updates while the USB2 PHY SUS Well is power gated. When set, the controller will block any updates to the PORTSC caused by port status change if the USB2 PHY SUS is power gated. 0 Do not<br><br>**Power Well:** SUS |
| 10:8 | 0x0<br>RW | **USB2 PHY SUS Well Power Gate Entry Hysteresis Count (U2PSPGEHC):** This controls the amount of hysteresis time the controller will enforce after detecting the USB2 PHY SUS Power Gate entry condition. 0h 0 clocks 1h 32 clocks 2h 64 clocks 3h 128 clocks 4h 256 clocks 5h 512 clocks 6h 1024 clocks 7h 2048 clocks<br><br>**Power Well:** SUS |
| 7:4 | 0x0<br>RW | **USB2 PHY SUS Power Gate PORTSC Block Policy (U2CLPGLAT):** This field represents the worst case latency for the USB2 Common Lane to enter and exit its power gate state. This fields is required to be compared to a ports HIRD/HIRD value for the ports that have allowed L1 to L2 mapping to determine if the Common Lane can be allowed to power off. If the power gate entry/exit latency is greater than the HIRD/HIRDD then the common lane should not be allowed to power gate as this will result in a L1 exit violation. 0h 100 us 1h 200 us 2h 300 us Eh 1500us Fh 1600 us<br><br>**Power Well:** SUS |
| 3:2 | 0x0<br>RW | **USB2 PHY SUS Well Power Gate Policy (U2PSUSPGP):** This field controls when to enable the USB2 PHY SUS Well Power Gating when the proper conditions are met. 00 USB2 PHY SUS Power Gating is Disabled. 01 USB2 PHY SUS Power Gating is Enabled in Only D0 and D0i2 (Excludes D0i3 and D3) 10 USB2 PHY SUS Power Gating is Eanabled in only in D0, D0i2 and D0i3 (Excludes D3) 11 USB2 PHY SUS Power Gating is Eanabled in D0/D0i2/D0i3/D3<br><br>**Power Well:** SUS |
| 1 | 0x0<br>RW | **USB2 Common Lane Power Gating Enable During L1 to L2 Mapping for USB2 PHY Power Gating (U2CLPGEL1L2):** This field when set enables the controller to allow for the common lane power gating to be enabled when all ports are exposed as in L2 to the USB2 PHY while at least 1 port has been mapped to L2 from L1. This field alone does not guarantee power gating since the L1 HIRD/HIRDD Value must be compared with the PHYs power gate exit latency (U2CLPGLAT) held in this register to ensure that L1 exit is not violated. 0 USB2 Common Lane Power Gating is disabled when any port has been mapped from L1 to L2. 1 USB2 Common Lane Power Gating is allowed when any port has been mapped to L2 from L1 with the additional condition that the HIRD/HIRDD is greater than the PHYs Power Gate exit latency.<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW | **USB2 Data Lane L1 to L2 Mapping Enable for USB2 PHY Power Gating (U2DLL1L2ME):** This field when set enables the controller to map an L1 entry directly to L2 to allow the USB2 PHY to trigger its Autonomous Power Gating. The USB2 PHY will trigger PG only when in L2 since it does not fully understand the requirements for L1. 0 USB2 L1 to L2 mapping is disabled for all ports 1 USB2 L1 to L2 mapping is enabled for all ports<br><br>**Power Well:** SUS |

## 51.3.179 ECC_PARITY_ERROR_LOG_REG—Offset 83F8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 83F8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **COMMAND_PARITY_DETECTED:** Command Parity Error detected on received IOSF transaction<br><br>**Power Well:** Core |
| 30 | 0x0<br>RW/1C | **DATA_PARITY_DETECTED:** Data Parity Error detected on received IOSF transaction<br><br>**Power Well:** Core |
| 29 | 0x0<br>RW/1C | **ERROR_PRESENT_DETECTED:** Error present detected on received IOSF transaction<br><br>**Power Well:** Core |
| 28:26 | 0x0<br>RO | **RSVD:** Reserved<br><br>**Power Well:** Core |
| 25:21 | 0x0<br>RW/1C | **Correctable ECC Error Source RF (CORRECTABLE_ECC_ERROR_SOURCE_RF):** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which Correctable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data<br><br>**Power Well:** Core |
| 20:15 | 0x0<br>RW/1C | **CORRECTABLE_ECC_ERROR_SOURCE_PORT:** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00<br><br>**Power Well:** Core |
| 14:13 | 0x0<br>RW/1C | **CORRECTABLE_ECC_ERROR_SOURCE_LOG:** When an Correctable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Correctable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.<br><br>**Power Well:** Core |
| 12:8 | 0x0<br>RW/1C | **UNCORRECTABLE_ECC_ERROR_SOURCE_RF:** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which uncorrectable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data<br><br>**Power Well:** Core |
| 7:2 | 0x0<br>RW/1C | **UNCORRECTABLE_ECC_ERROR_SOURCE_PORT:** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1:0 | 0x0<br>RW/1C | **UNCORRECTABLE_ECC_ERROR_SOURCE_LOG:** When an uncorrectable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Uncorrectable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.<br><br>**Power Well:** Core |

## 51.3.180 ECC_POISONING_CTRL_REG—Offset 83FCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 83FCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RW | **ENGINE_RF_ECC_POISONING_VECTOR:** XHCI Engine RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width<br><br>**Power Well:** Core |
| 22:14 | 0x0<br>RW | **USB3_RF_ECC_POISONING_VECTOR:** USB3 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width<br><br>**Power Well:** Core |
| 13:5 | 0x0<br>RW | **USB2_RF_ECC_POISONING_VECTOR:** USB2 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width<br><br>**Power Well:** Core |
| 4:3 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **ENGINE_RF_ECC_POISONING_EN:** Enable ECC Poisoning for XHCI Engine related RFs that support ECC<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **USB3_RF_ECC_POISONING_EN:** Enable ECC Poisoning for USB3 Port related RFs that support ECC. Setting this bit enables poisoning of USB3 Port related RFs. This applies to all USB3 ports<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **USB2_RF_ECC_POISONING_EN:** Enable ECC Poisoning for USB2 Port related RFs that support ECC. Setting this bit enables poisoning of USB2 Port related RFs. This applies to all USB2 ports<br><br>**Power Well:** Core |

## 51.3.181 USB2_PORT_STATE_REG—Offset 8400h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MBAR] + 8400h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:4 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** SUS |
| 3:0 | 0x0<br>RO | **USB2_PORT_STATE_REG:** Per USB2 Port State Register indicating the following states 0x0 Connected 0x1 Suspended 0x2 Disabled 0x3 Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.<br><br>**Power Well:** SUS |

## 51.3.182 USB2_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MBAR] + 8408h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:4 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** SUS |
| 3:0 | 0x0<br>RO | **USB3_PORT_STATE_REG:** Per USB3 Port State Register indicating the following states 0x0 Connected 0x1 Suspended 0x2 Disabled 0x3 Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.<br><br>**Power Well:** SUS |

## 51.3.183 FUS1_REG—Offset 8410h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00010000h

**BAR and Offset:** [MBAR] + 8410h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 20:18 | 0x0<br>RO | **DEV_ID:** Provides a fuse over-ride for the lower 3 bits of device ID.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RO | **XHC_DPGDIS:** When asserted, it indicates that the xHCI will not dynamically power gate the controller.<br><br>**Power Well:** Core |
| 16 | 0x1<br>RO | **USBR_DIS:** When asserted, it indicates that xHCI does not support USBr<br><br>**Power Well:** Core |
| 15 | 0x0<br>RO | **XHC_FXN_DIS:** When asserted, it indicates the xHCI is fused to function disabled.<br><br>**Power Well:** Core |
| 14:10 | 0x0<br>RO | **USB2_PORT_COUNT:** 0x0 = MAX USB2 Ports enabled 0x1 = Max USB2 Ports -1 enabled 0x1F = All ports disabled<br><br>**Power Well:** Core |
| 9:5 | 0x0<br>RO | **USB3_PORT_COUNT:** 0x0 = MAX USB3 Ports enabled 0x1 = Max USB3 Ports -1 enabled 0x1F = All ports disabled<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **DEBUG_MODE_ENABLE:** 0 = USB debug mode disabled 1 = USB debug mode enabled Not used for LPT.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **XHC_DCGDIS:** 0= USB3 (xHC) dynamic clock gating enabled 1= USB3 (xHC) dynamic clock gating disabled<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **REGFILE_DPGDIS:** 0 = USB3 (xHC) dynamic RF power gating enabled 1 = USB3 (xHC) dynamic RF power gating disabled<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO | **USBIO_PMDIS:** 0=USB2 HW LPM and USB3 HW Ux under xHC enabled 1= USB2 HW LPM and USB3 HW Ux under xHC disabled<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **USB2_PLL_SHUTDOWN_DIS:** 0= USB2 PLL Shutdown enabled 1= USB2 PLL Shutdown disabled<br><br>**Power Well:** Core |

## 51.3.184 FUS2_REG—Offset 8414h

This register is NOT subject to HW save and restore.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8414h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 11:0 | 0x0<br>RO | **PORT_MAP_FUS1:** 1:0 Port 1 3:2 Port 2 5:4 Port 3 31:30 Port 16 Bit Description: 00: Port assgned to USB3 01: Port assigned to non-XHCI controller 10: Port assigned to Flex IO mapping. Mapping is based on Soft Straps. 11: Reserved.<br><br>**Power Well:** SUS |

## 51.3.185 STRAP1_REG—Offset 841Ch

This register is NOT subject to HW save and restore.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 841Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **FlexIO mapping (FLEX_IO_MAPPING):** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is owned by XHCI 1: Port is not owned by XHCI<br><br>**Power Well:** Core |

## 51.3.186 STRAP2_REG—Offset 8420h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8420h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 5:0 | 0x0<br>RO | **USB3_SSIC_MODE:** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is operated in USB3 mode 1: Port is operated in SSIC mode.<br><br>**Power Well:** Core |

## 51.3.187 STRAP3_REG—Offset 8424h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8424h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RO | **XHCI_PRI_CLK_FREQ_SEL:** Specifies the frequency of the Primary clock (iosf_prim_mux_clk) used by XHCI 00 : 200 MHz (default) 01 : 125 MHz 10 : 250 MHz<br><br>**Power Well:** SUS |

## 51.3.188 DFT_REG1—Offset 8430h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8430h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **SS/SSIC DFT CRC (SSDFTCRC):** These register bits contain the value of SS DFT CRC<br><br>**Power Well:** Core |
| 15:12 | 0x0<br>RO | **RSVD0:** RSVD0<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **DFTLFPSSEL:** DFTLFPSSEL 0: Rxelecidle is driven from GPIO Pin 1: Rxelecidle is Driven as specified by Super Speed DFT LFPS Mode Select<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **Super Speed DFT LFPS Mode Select (SSDFTLMSEL):** This bit selects the Super Speed DFT LFPS mode, and will only take effect when Super Speed HBP mode is enabled. 0b: HBP logic will internally loopback TX LFPS as RX LFPS and AFE RX LFPS path to the controller is disconnected. 1b: RX LFPS path works normally<br><br>**Power Well:** Core |
| 9:6 | 0x0<br>RW | **SS/SSIC DFT CRC Select (SSDFTCRCSEL):** These bits select which Super Speed DFT CRC value is reflected in SSDFTCRC bits. In addition, these bits also select which SuperSpeed DFT CRC MSB value is sent to GPIO monitor pin, i.e. sata3gp_gp37 to aid silicon debug. Live version of selected CRC's MSB will be- sent to GPIO monitor pin, i.e. sata3gp_gp37. 0h (default): No SuperSpeed DFT CRC is selected. 1h: Data Payload CRC 2h: Link Management Packet CRC 3h: Transaction Packet CRC 4h: Isochronous Timestamp Packet CRC 5h: Data Packet Header CRC 6h: Link Command Packet CRC 7h: RRAP Packet CRC 8h: Tx/Rx Cfg CRC Others: Reserved Note : CRC types 7 and 8 are onlyapplicable if Port is SSIC Port<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **RSVD1:** RSVD1<br><br>**Power Well:** Core |
| 4:0 | 0x0<br>RW | **SS/SSIC DFT CRC Port Select (SSICDFTCPS):** One CRC per packet type is shared for all the Super Speed ports. These bits select the Super Speed port for which CRC data will be updated. 000b: (default) No SuperSpeed Port is selected 001b: SS/SSIC Port 0 010b: SS/SSIC Port 1 011b: SS/SSIC Port 2 100b: SS/SSIC Port 3 101b: SS/SSIC PORT4 110b :SS/SSIC Port 5 others : Rsvd<br><br>**Power Well:** Core |

## 51.3.189 DFT_REG2—Offset 8434h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8434h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **HS/HSIC TX CRC (TXCRC):** These register bits contain the value of TX CRC. This TX CRC is placed right after the 480MHz XCLKQ.<br><br>**Power Well:** Core |
| 15:11 | 0x0<br>RW | **HSIC/UTMI+ DFT Port Select (UTMIDFTPS):** One CRC is shared for all the UTMI+ ports. These bits select the UTMI+ port for which CRC data will be updated and loopback status reflected in UTMILPBKSTS. 0h: (default) No UTMI+ Port is selected 1h: UTMI+ Port 0 2h: UTMI+ Port 1 3h: UTMI+ Port 2 4h: UTMI+ Port 3 5h: UTMI+ Port 4 6h: UTMI+ Port 5 7h: UTMI+ Port 6 8h: UTMI+ Port 7 9h: UTMI+ Port 8 Ah: UTMI+ Port 9 Bh: UTMI+ Port 10 Ch: UTMI+ Port 11 Dh: UTMI+ Port 12 Eh: UTMI+ Port 13 Others: Reserved<br><br>**Power Well:** Core |
| 10:7 | 0x0<br>RW | **Loop Number (UTMILPBKLOOPN_3_0):** Number of repeatable fixed pattern within a packet Note: Connect register bit 3 to counter bit 7, register bit 2 to counter bit 5, register bit 1 to counter bit 3, register bit 0 to counter bit 1. Counter bits 6, 4, 2 and 0 will be tied off to 0. Hence, the programmable loop number shall be: 0000b: 0 loop 0001b: 2 loops 0010b: 8 loops 0011b: 10 loops 1100b: 160 loops 1101b: 162 loops 1110b: 168 loops 1111b: 170 loops (max) +E27<br><br>**Power Well:** Core |
| 6:5 | 0x0<br>RW | **Operational Mode (UTMIOPMODE_1_0):** Operational Mode in test mode. These signals select between various operational modes: 00b: Normal Operation 01b: Non-Driving 10b: Disable Bit Stuffing and NRZI encoding 11b: Reserved<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW | **Termination Select (UTMITERMSEL):** Termination Select in test mode. This signal selects between the FS and HS terminations: 0b: HS termination enabled 1b: FS termination enabled<br><br>**Power Well:** Core |
| 3:2 | 0x0<br>RW | **Transceiver Select (UTMIXCVRSELECT_1_0):** Transceiver Select in test mode. This signal selects between the LS, FS and HS transceivers: 00b: HS transceiver enabled 01b: FS transceiver enabled 10b: LS transceiver enabled 11b: Reserved<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RW | **UTMI+ Loopback Status (UTMILPBKSTS):** Loopback Status for port selected by UTMIDFTPS 00b: Reset condition 01b: Comparator has started receiving data and the received data matches with the TX pattern. 10b: Comparator has started receiving data the received data does not match with the TX pattern but there was no assertion of RC error from UTMI. 11b: Comparator has started receiving data and RX ERROR was asserted for at least one clock by UTMI. Note that this does not reflect the status of pattern comparison since RX error from UTMI is unexpected for loopback.<br><br>**Power Well:** Core |

## 51.3.190 DFT_REG3—Offset 8438h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8438h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Loopback Pattern (UTMILPBKPAT):** 2-byte pattern for loopback Note: This 2-byte pattern will be replicated to the upper 2-byte to form the DW pattern<br><br>**Power Well:** Core |
| 15:2 | 0x0<br>RW | **Loopback Lane Select (UTMILPBKSEL_13_0):** Port is selected if the corresponding bit is selected Note: MSB is for UTMI+ Port13, LSB is for UTMI+ Port0<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Loopback Pattern (UTMILPBKPATSEL):** Near end loopback pattern generation type: 0b: Fixed pattern 1b: USB2 test packet<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Loopback Type (UTMILPBKTYPE):** Loopback Type in test mode. This signal selects between DNELB and ANELB, and will only take effect if DTUTMILPBKEN is set. 0b: Digital Near-End Loopback (DNELB) 1b: Analog Near-End Loopback (ANELB) Note: Analog Far-End Loopback (AFELB) is not supported<br><br>**Power Well:** Core |

## 51.3.191 dft_reg4 (DFT_REG4)—Offset 843Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 843Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:11 | 0x0<br>RW | **RSVD:** Reserved.<br><br>**Power Well:** Core |
| 10:0 | 0x0<br>RW | **Loopback Pattern (UTMILPBKPATSEL):** Loopback Lane Select (UTMILPBKSEL): Port is selected if the corresponding bit (zero based) is selected Bit 0 = Port 1 Bit 1 = Port 2 Etc.<br><br>**Power Well:** Core |

## 51.3.192 dft_reg5 (DFT_REG5)—Offset 8440h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8440h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RW/L | **High Speed Bypass Enable (HIGH_SPEED_BYPASS_ENABLE):** 0000 - Disable all HBP 0001 - Enable SS HBP 0010 - Enable HS HBP 0011 - Enable HSIC HBP 0100 - Enable SSIC HBP Others - Reserved<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **DFTIDPINSEL (DFTIDPINSEL):** Select Between Device and Host Controller in HBP mode 0h : Device Controller is selected 1h : Host Controller is selected<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW/O | **Lock Bit (LOCK):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to HBP Enable is disabled (locked state). When set to '0', the write access to bit locked are enable (unlocked state). Writable once after platform reset.<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW/L | **Loopback Enable (UTMILPBKEN):** Enable loopback test mode. If asserted, loopback test mode is enabled<br><br>**Power Well:** Core |
| 24:0 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |

## 51.3.193 XECP_CMDM_STS0—Offset 8448h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8448h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **IDMA_OWNS_CNTX:** Indicates that IDMA module owns the context access currently<br><br>**Power Well:** Core |
| 30 | 0x0<br>RO | **ODMA_OWNS_CNTX:** Indicates that ODMA module owns the context access currently<br><br>**Power Well:** Core |
| 29 | 0x0<br>RO | **TRM_OWNS_CNTX:** Indicates that TRM modules owns the context access currently<br><br>**Power Well:** Core |
| 28 | 0x0<br>RO | **CMD_RING_REQUESTED_CNTX_LOCK:** Indicates that Command Manager has requested a context lock<br><br>**Power Well:** Core |
| 27 | 0x0<br>RO | **CMD_RING_STOP_IN_PROGRESS:** Indicates that Command Ring stop command is in progress<br><br>**Power Well:** Core |
| 26 | 0x0<br>RO | **SCH_UPDATE_CLR_EP_IN_PROGRESS:** Indicates that clearing an EP out of schedule is in progress<br><br>**Power Well:** Core |
| 25 | 0x0<br>RO | **ADDR_DEV_DONE:** Indicates that current address device command is done by ODMA<br><br>**Power Well:** Core |
| 24 | 0x0<br>RO | **ADDR_DEV_IN_PROGRESS:** Indicates that ODMA has an address device command in progress<br><br>**Power Well:** Core |
| 23 | 0x0<br>RO | **EP_STATE_UPDATE_IN_PROGRESS:** Indicates that updating of EP state is in progress<br><br>**Power Well:** Core |
| 22 | 0x0<br>RO | **EP_STATE_IN_PROGRESS_FROM_STOP_DUE_TO_DB:** Indicates that doorbell manager is issuing and EP update due to a doorbell ring on an EP that is in stop state<br><br>**Power Well:** Core |
| 21 | 0x0<br>RO | **EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_EP_ERROR:** Indicates that transfer ring manager is isuing and EP update due to an EP error condition detected<br><br>**Power Well:** Core |
| 20 | 0x0<br>RO | **EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_STALL:** Indicates that transfere ring manager is issuing an EP state update due to stall received<br><br>**Power Well:** Core |
| 19 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RO | **STOP in progress (STOP_IN_PROGRESS):** Indicates that a STOP on the Command Ring is in progress<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **commend ring has doorbell pending (CMD_RING_DB_PENDING):** Indicates that the commend ring has doorbell pending<br><br>**Power Well:** Core |
| 16 | 0x0<br>RO | **commend ring running (CMD_RING_RUNNING):** Indicates that the commend ring is running<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Command next capability offset (CMD_NEXT_CAP_OFFSET):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RO | **Vendor defined capability ID (VID):** Reserved.<br><br>**Power Well:** Core |

## 51.3.194 XECP_CMDM_STS2—Offset 8450h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8450h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **Rsvd0:** Reserved.<br><br>**Power Well:** Core |
| 4:0 | 0x0<br>RO | **Event Ring Segment Table (ERST):** count low<br><br>**Power Well:** Core |

## 51.3.195 XECP_CMDM_STS3—Offset 8454h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8454h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Rsvd0:** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Event Ring Segment Table (ERST):** count high<br><br>**Power Well:** Core |

## 51.3.196 AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8460h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **RESERVED (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>WO | **Allow Software USB PHY RST (ALL_SW_UP_RST):** Allow a USB PHY reset being issued by software. Writing to this register with bit set to 1 will reset the USB PHY that is connected to the port. Bit3:0 indicates the port number of the USB PHY<br><br>**Power Well:** SUS |

## 51.3.197 Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h

The Latency Tolerance Control Register is used by SW to control which BELT is returned when this register is read. SW shall write to this register to program a Slot-ID, Port-ID and BELT Select to determine which BELT is selected. When this register is read the selected BELT is returned.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 8464h

**Default:** 000D0000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>WO | **BELT Select (BELT_SEL):** This field determines what value will be selected to read back from SW when reading this register 0: Returns the SW programmed Latency Tolerance Value 1: Returns the Lowest BELT in the Host 2: Returns the BELT for the requested Slot-ID (Slot Select) 3: Returns the BELT for the requested Port-ID (Port Select)<br><br>**Power Well:** Core |
| 29:20 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 19:16 | 0xd<br>WO | **Port Select (PORT_SEL):** Used to select the BELT for a given Port # when the BELT Select is programmed to select the Port-ID (this field is 0 based)<br><br>**Power Well:** Core |
| 15:12 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 11:5 | 0x0<br>RO | **BELT Value (BELTV):** Value of selected BELT is return in this field<br><br>**Power Well:** Core |
| 4:0 | 0x0<br>RW | **Slot Select (SLOT_SEL):** Reads will return: BELT Value (BELTV) [4:0]: Value of selected BELT is return in this field Writes will control : Slot Select (): Used to select the BELT for a given Slot # when the BELT Select is programmed to select the Slot-ID (this field is zero based)<br><br>**Power Well:** Core |

## 51.3.198 USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00002201h

**BAR and Offset:** [MBAR] + 846Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | Reserved. |
| 24 | 0x0<br>RW | **HC OS Owned Semaphore (HCOSOS):** Default = '0'. System software sets this bit to request ownership of the xHC. Ownership is obtained when this bit reads as '1' and the HC BIOS Owned Semaphore bit reads as '0'. |
| 23:17 | 0x0<br>RO | Reserved. |
| 16 | 0x0<br>RW | **HC BIOS Owned Semaphore (HCBIOSOS):** Default = '0'. The BIOS sets this bit to establish ownership of the xHC. System BIOS will set this bit to a '0' in response to a request for ownership of the xHC by system software. |
| 15:8 | 0x22<br>RW/S | **Next Capability Pointer (NextCP):** This field indicates the location of the next capability with respect to the effective address of this capacity. Refer to Table 145 for more information on this field. |
| 7:0 | 0x1<br>RW/L | **Capability ID (CID):** This field identifies the extended capability. Refer to Table 146 for the value that identifies the capability as Legacy Support. This extended capability requires one additional 32-bit register for control/status information (USBLEGCTLSTS), and this register is located at offset xECP+04h. |

## 51.3.199 USB2 Port Disable Override (USB2PDO)—Offset 84F8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 84F8h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | Reserved. |
| 13:0 | 0x0<br>RW/O | **USB2PDO (USB2PDO):** A '1' in a bit position prevents the corresponding USB 2 port from reporting a Device Connection to the xHC. This applies across all USB2 protocol ports<br>• 0 - Allows corresponding USB port to report a device connection to the xHC.<br>• 1 - Prevents the corresponding USB port from reporting a device Connection to the xHC.<br>Port to bit mapping is in one-hot encoding, this is bit 0 controls port 1 and so on.<br>• Bit 0 - USB 2.0 port 0<br>• ...<br>• Bit N - USB 2.0 port N |

## 51.3.200 USB3 Port Disable Override (USB3PDO)—Offset 84FCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 84FCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | Reserved. |
| 9:0 | 0x0<br>RW/O | **USB3 Port Disable Override (USB3PDO):**<br>• 0 - Allows corresponding USB port to report a Device Connection to the xHC.<br>• 1 - Prevents the corresponding USB port from reporting a Device Connection to the xHC.<br>• Bit 0 - USB 3.0 Port 1<br>• ...<br>• Bit N - USB 3.0 Port N |

## 51.3.201 Command (CMD_MMIO)—Offset 8604h

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [MBAR] + 8604h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller. Note that the corresponding Interrupt Status bit is not affected by the interrupt enable.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBE):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **SERR# Enable (SERR):** When set to 1, the XHC is capable of generating (internally) SERR#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Wait Cycle Control (WCC):** Reserved.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification. Note that this applies to both requests and completions from the system interface. This bit must be set in order for the parity errors to generate SERR#.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VPS):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Memory Write Invalidate (MWI):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** When set, it allows XHC to act as a bus master. When cleared, it disable XHC from initiating transactions on the system bus.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers. If this bit is set, accesses to the XHC registers are enabled. The Base Address register for the XHC should be programmed before this bit is set.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |

## 51.3.202 Device Status (STS_MMIO)—Offset 8606h

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0290h

**BAR and Offset:** [MBAR] + 8606h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **Detected Parity Error (DPE):** This bit is set by the SoC whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW/1C | **Signaled System Error (SSE):** This bit is set by the SoC whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW/1C | **Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW/1C | **Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW/1C | **Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.<br><br>**Power Well:** Core |
| 10:9 | 0x1<br>RO | **DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. Read-Only.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW/1C | **Master Data Parity Error Detected (MDPED):** This bit is set by the SoC whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a 1 to this bit location.<br><br>**Power Well:** Core |
| 7 | 0x1<br>RO | **Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RO | **User Definable Features (UDF):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **66 MHz Capable (MC):** Reserved as 0. Read-Only.<br><br>**Power Well:** Core |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. This bit is a 1 when the interrupt is asserted. This bit will be 0 when the interrupt is deasserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 51.3.203 Revision ID (RID_MMIO)—Offset 8608h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 8608h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):**<br><br>**Power Well:** Core |

## 51.3.204 Programming Interface (PI_MMIO)—Offset 8609h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 30h

**BAR and Offset:** [MBAR] + 8609h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x30<br>RO | **Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.<br><br>**Power Well:** Core |

## 51.3.205 Sub Class Code (SCC_MMIO)—Offset 860Ah

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 03h

**BAR and Offset:** [MBAR] + 860Ah

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x3<br>RO | **Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.<br><br>**Power Well:** Core |

## 51.3.206 Base Class Code (BCC_MMIO)—Offset 860Bh

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 0Ch

**BAR and Offset:** [MBAR] + 860Bh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0xc<br>RO | **Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.<br><br>**Power Well:** Core |

## 51.3.207 Master Latency Timer (MLT_MMIO)—Offset 860Dh

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 860Dh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.<br><br>**Power Well:** Core |

## 51.3.208 Header Type (HT_MMIO)—Offset 860Eh

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 860Eh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Multi-Function Bit (MFB):** Read only indicating single function device.<br><br>**Power Well:** Core |
| 6:0 | 0x0<br>RO | **Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.<br><br>**Power Well:** Core |

## 51.3.209 Memory Base Address (MBAR_MMIO)—Offset 8610h

Value in this register will be different after the enumeration process.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000004h

**BAR and Offset:** [MBAR] + 8610h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:16 | 0x0<br>RW | **Base Address (BA):** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.<br><br>**Power Well:** Core |
| 15:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. Read-Only 0, this indicates that this function is requesting an 64KB block of memory.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Prefetchable:** This bit is hardwired to 0 indicating that this range should not be prefetched.<br><br>**Power Well:** Core |
| 2:1 | 0x2<br>RO | **Type:** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space. If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space<br><br>**Power Well:** Core |

## 51.3.210 USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [MBAR] + 862Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.<br><br>**Power Well:** Core |

## 51.3.211 USB Subsystem ID (SSID_MMIO)—Offset 862Eh

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [MBAR] + 862Eh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/L | **USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).<br><br>**Power Well:** Core |

## 51.3.212 Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 70h

**BAR and Offset:** [MBAR] + 8634h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x70<br>RO | **Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.<br><br>**Power Well:** Core |

## 51.3.213 Interrupt Line (ILINE_MMIO)—Offset 863Ch

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 863Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** This data is not used by the SoC. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.<br><br>**Power Well:** Core |

## 51.3.214 Interrupt Pin (IPIN_MMIO)—Offset 863Dh

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 863Dh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/L | **Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).<br><br>**Power Well:** Core |

## 51.3.215 XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000001FDh

**BAR and Offset:** [MBAR] + 8640h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/O | **Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to bits locked by this bit is disabled (locked state). When set to '0', the write access to bit locked by this bit is enabled (unlocked state). Writable once after platform reset.<br><br>**Power Well:** Core |
| 30:25 | 0x0<br>RW | **ECO1:**<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.<br><br>**Power Well:** Core |
| 23 | 0x0<br>RW/C | **Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle. Cleared by SW when the bit is written with value of '1'.<br><br>**Power Well:** Core |
| 22 | 0x0<br>RW | **Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.<br><br>**Power Well:** Core |
| 21:19 | 0x0<br>RW | **Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified. 000: Disabled 001: 32 bb_cclk 010: 64 bb_cclk 011: 128 bb_cclk 100: 256 bb_cclk 101: 512 bb_cclk 110: 1024 bb_cclk 111: 131072 bb_cclk<br><br>**Power Well:** Core |
| 18 | 0x0<br>RW | **XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power mangement to be enabled.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW | **D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power managment to be enables. This bit can only be set if the XHCI L1 Override P2 chicken bit is set.<br><br>**Power Well:** Core |
| 16:12 | 0x0<br>RW | **Periodic Complete Pre Wake Time (PCPWT):** signal . This allows for platform wake time before the next scheduled periodic transaction. The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert. This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame. Register Format: Bits (16:12) represnets the # of bytes remaining with a 256B granularity. Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated. This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected. This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x1<br>RW | **L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC. 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk<br><br>**Power Well:** Core |
| 7:6 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type<br><br>**Power Well:** Core |
| 5:4 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type<br><br>**Power Well:** Core |
| 3:2 | 0x3<br>RW | **Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type<br><br>**Power Well:** Core |
| 1:0 | 0x1<br>RW | **Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter<br><br>**Power Well:** Core |

## 51.3.216 XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0003C000h

**BAR and Offset:** [MBAR] + 8644h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration. It must be set by BIOS after configuration of the OC mapping bits is complete. Once this bit is set, OC mapping shall not be changed by SW.<br><br>**Power Well:** SUS |
| 30:26 | 0x0<br>RW | **ECO1:**<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request.<br>• 0: 4KB<br>• 1: 64B<br><br>**Power Well:** Core |
| 24:22 | 0x0<br>RW | **IDMA Write Request Size Control (IDMA_WRREQSZCTRL):** Write Request Size Control:<br>This bit controls the maximum size of each Write Request.<br>• 000: 128B<br>• 001: 256B<br>• 011 - 110: Reserved<br>• 111: 64B<br><br>**Power Well:** Core |
| 21 | 0x0<br>RW | **XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads.<br>• 0 - xHC will clear RO for all upstream read requests.<br>• 1 - xHC will set RO for all upstream read requests.<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW | **IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.<br><br>**Power Well:** Core |
| 19:14 | 0xf<br>RW | **Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource.<br>This value is zero based.<br>000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation If set less than default allows under-allocation Only allowed to be programmed when BME = 0 and no outstanding downstream completion<br><br>**Power Well:** Core |
| 13:12 | 0x0<br>RW | **SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit.<br>• 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state<br>• 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW.<br>• 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI.<br>• 11b: Reserved<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x0<br>RW | **MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write.<br>• 0b (Default): Delay MMIO Read after MMIO Write<br>• 1b: Do not delay MMIO Read after MMIO Write<br>***Note:*** This delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write.<br>• 0b (Default): Do not delay MMIO Write after previous MMIO Write<br>• 1b: Delay MMIO Write after previous MMIO Write<br>***Note:*** The delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.<br><br>**Power Well:** Core |
| 9:8 | 0x0<br>RW | **SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave.<br>• 00: Never inhibit Cx<br>• 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior)<br>• 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1<br>• 11: Always inhibit Cx<br><br>**Power Well:** Core |
| 7:6 | 0x0<br>RW | **SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave.<br>• 00: Never inhibit DMI L1.<br>• 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior).<br>• 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1.<br>• 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.<br><br>**Power Well:** Core |
| 5:3 | 0x0<br>RW | **L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified. If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake,<br>• 000: Disabled<br>• 001: 128 bb_cclk<br>• 010: 256 bb_cclk<br>• 011: 512 bb_cclk<br>• 100: 1024 bb_cclk<br>• 101: 2048 bb_cclk<br>• 110: 4096 bb_cclk<br>• 111: 131072 bb_cclk<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Read Request Size Control (RDREQSZCTRL):** Read Request Size Control:<br>This bit controls the maximum size of each Read Request.<br>• 000: 128B<br>• 001: 256B<br>• 010: 512B<br>• 011 - 110: Reserved<br>• 111: 64B<br><br>**Power Well:** Core |

## 51.3.217 Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8650h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **Rsvd2:** Reserved<br><br>**Power Well:** Core |
| 28 | 0x0<br>RW | **Naking USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not. 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation. 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.<br><br>**Power Well:** Core |
| 27 | 0x0<br>RW | **SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met 0 - Disallow SRAM Power Gating. 1 - Allow SRAM Power Gating<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW | **SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports ontop of trunk clock gating. 0 - P3 state NOT allowed to result in PXP PLL shutdown. 1- P3 state allowed to result in PXP PLL shutdown<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request. Note: if USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.<br><br>**Power Well:** Core |
| 23:20 | 0x0<br>RW | **HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) NA (no support for U1) (2) ==) U2 (L1) or deeper (3) ==) U3 (L2) or deeper<br><br>**Power Well:** Core |
| 19:16 | 0x0<br>RW | **SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating. 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state. 1 - xHC may allow frame timer to be gated regardless of EU3S.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14 | 0x0<br>RW | **XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated. 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running. 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB. Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock. 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle. 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW | **XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is atleast 1 non Remote Wake Enabled HS Port in U2. 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2. 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.<br><br>**Power Well:** Core |
| 11:10 | 0x0<br>RW | **XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports. Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted. (0) ==) L1 or deeper (1) ==) L2 or deeper<br><br>**Power Well:** Core |
| 9:8 | 0x0<br>RW | **HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions. PLL Shutdown is allowed in: 00b Disabled (Link states shall be disabled for DMI PLL shutdown) 01b U0 or conditions for 10b setting. 10b U2 or conditions for 11b setting. 10b U3, Disconnected, Disabled or Powered-Off.<br><br>**Power Well:** Core |
| 7:5 | 0x0<br>RW | **SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions. PLL Shutdown is allowed in: 000b Disabled (Link states shall be ignored for DMI PLL shutdown). 001b U0 or conditions for 010b setting 010b U1 or conditions for 011b setting 011b U2 or conditions for 100b setting 100b U3, Disconnected, Disabled or Powered-Off<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW | **XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.<br><br>**Power Well:** Core |

## 51.3.218 Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 8658h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Rsvd2:** Reserved.<br><br>**Power Well:** Core |
| 29:16 | 0x0<br>RO/V | **Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits (29:16) reflect state of bits (13:0) of FRINDEX<br><br>**Power Well:** Core |
| 15:13 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RO/V | **Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.<br><br>**Power Well:** Core |

## 51.3.219 Serial Bus Release Number (SBRN_MMIO)—Offset 8660h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [MBAR] + 8660h

**Default:** 30h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x30<br>RO | **Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.<br><br>**Power Well:** SUS |

## 51.3.220 Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 60h

**BAR and Offset:** [MBAR] + 8661h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):** .<br><br>**Power Well:** SUS |
| 6 | 0x1<br>RO | **No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.<br><br>**Power Well:** Core |
| 5:0 | 0x20<br>RO | **Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal) 59488 0 (00h) 59504 1 (01h) 59520 2 (02h) ... 59984 31 (1Fh) 60000 32 (20h) ... 60480 62 (3Eh) 60496 63 (3Fh) Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value<br><br>**Power Well:** SUS |

## 51.3.221 Best Effort Service Latency (BESL_MMIO)—Offset 8662h

Bset Effort Service Latency.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 8662h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RW/L | **Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field. This is programmed by BIOS based on platform parameters.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW/L | **Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field. This is programmed by BIOS based on platform parameters.<br><br>**Power Well:** Core |

## 51.3.222 PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 01h

**BAR and Offset:** [MBAR] + 8670h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RO | **PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.<br><br>**Power Well:** Core |

## 51.3.223 Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 80h

**BAR and Offset:** [MBAR] + 8671h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x80<br>RW/L | **Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h. This register is writable when the Access Control bit is set to '0'. This allows BIOS to effectively hide the next capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden. Note: This value is never expected to be programmed.<br><br>**Power Well:** Core |

## 51.3.224 Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h

Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the SoC is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [MBAR] + 8672h

**Default:** C1C2h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x18<br>RW/L | **PME_Support:** This 5-bit field indicates the power states in which the function may assert PME#. The SoC XHC does not support the D1 or D2 states. For all other states, the SoC XHC is capable of generating PME#. Software should never need to modify this field.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW/L | **D2_Support:** The D2 state is not supported.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RW/L | **D1_Support:** The D1 state is not supported.<br><br>**Power Well:** Core |
| 8:6 | 0x7<br>RW/L | **Aux_Current:** The SoC XHC reports 375mA maximum Suspend well current required when in the D3cold state. This value can be written by BIOS when a more accurate value is known.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW/L | **DSI:** The SoC reports 0, indicating that no device-specific initialization is required.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/L | **PME Clock (PMEClock):** The SoC reports 0, indicating that no PCI clock is required to generate PME#.<br><br>**Power Well:** Core |
| 2:0 | 0x2<br>RW/L | **Version:** The SoC reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.<br><br>**Power Well:** Core |

## 51.3.225 Power Management Control/Status (PM_CS_MMIO)—Offset 8674h

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0008h

**BAR and Offset:** [MBAR] + 8674h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **PME_Status:** This bit is set when the SoC XHC would normally assert the PME# signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). Writing a 0 has no effect. This bit must be explicitly cleared by the operating system each time the operating system is loaded.<br><br>**Power Well:** SUS |
| 14:13 | 0x0<br>RO | **Data_Scale:** The SoC hardwires these bits to 00 because it does not support the associated Data register.<br><br>**Power Well:** Core |
| 12:9 | 0x0<br>RO | **Data_Select:** The SoC hardwires these bits to 0000 because it does not support the associated Data register.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **PME_En:** A 1 enables the SoC XHC to generate an internal PME signal when PME_Status is 1. This bit must be explicitly cleared by the operating system each time it is initially loaded.<br><br>**Power Well:** SUS |
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** ), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. Transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **Reserved (RSVD2):** Reserved.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RW | **PowerState:** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state. The definition of the field values are: 00b - D0 state 11b - D3hot state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3hot state, the SoC must not accept accesses to the XHC memory range, but the configuration space must still be accessible.<br><br>**Power Well:** Core |

## 51.3.226 Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 05h

**BAR and Offset:** [MBAR] + 8680h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates that this is an MSI capability<br><br>**Power Well:** Core |

## 51.3.227 Next item pointer (MSI_NEXT_MMIO)—Offset 8681h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MBAR] + 8681h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates that this is the last item on the capability list<br><br>**Power Well:** Core |

## 51.3.228 Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0086h

**BAR and Offset:** [MBAR] + 8682h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RO | **Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported<br><br>**Power Well:** Core |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RW | **Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.<br><br>**Power Well:** Core |
| 3:1 | 0x3<br>RO | **Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert. This field must be set by HW to reflect the number of Interrupters supported. Encoding number of Vectors requested (number of Interrupters) 000 1 001 2 010 4 011 8 100 16 101 32 110-111 Reserved<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts. If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.<br><br>**Power Well:** Core |

## 51.3.229 Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8684h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Addr:** Lower DW of system specified message address, always DWORD aligned<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 51.3.230 Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8688h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Addr (UpperAddr):** Upper DW of system specified message address.<br><br>**Power Well:** Core |

## 51.3.231 Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [MBAR] + 868Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction. The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000, the function is not permitted to modify the message data.<br><br>**Power Well:** Core |

## 51.3.232 Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** F0140009h

**BAR and Offset:** [MBAR] + 8690h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0xf<br>RO | **VID:** Reserved.<br><br>**Power Well:** Core |
| 27:24 | 0x0<br>RO | **REV:** Reserved.<br><br>**Power Well:** Core |
| 23:16 | 0x14<br>RO | **Length (LENGTH):** Indicates that this capability is 16 bytes long.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.<br><br>**Power Well:** Core |
| 7:0 | 0x9<br>RO | **Capability ID (CID):** Reserved.<br><br>**Power Well:** Core |

## 51.3.233 Vendor Specific Header (VSHDR_MMIO)—Offset 8694h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 01400010h

**BAR and Offset:** [MBAR] + 8694h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x14<br>RO | **VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register<br><br>**Power Well:** Core |
| 19:16 | 0x0<br>RO | **VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.<br><br>**Power Well:** Core |
| 15:0 | 0x10<br>RO | **VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.<br><br>**Power Well:** Core |

## 51.3.234 SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8698h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a dont care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 3:1 | 0x0<br>RO | **Base Address Register Number (BARNUM):** Contains the 0s based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a dont care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Valid (VALID):** Set to 1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to 0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.<br><br>**Power Well:** Core |

## 51.3.235 Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00080AC1h

**BAR and Offset:** [MBAR] + 869Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x80ac<br>RW/L | **DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a dont care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 3:1 | 0x0<br>RO | **Base Address Register Number (BARNUM):** Contains the 0s based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a dont care, if the Valid bit is not set.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RW/L | **Valid (VALID):** Set to 1 to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to 0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.<br><br>**Power Well:** Core |

## 51.3.236 Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0800h

**BAR and Offset:** [MBAR] + 86A0h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 12:10 | 0x2<br>RW/L | **Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is 0.<br><br>**Power Well:** Core |
| 9:0 | 0x0<br>RW/L | **Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is 0.<br><br>**Power Well:** Core |

## 51.3.237 High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00002000h

**BAR and Offset:** [MBAR] + 86A4h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Rsvd1:** Reserved.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RW | **PORT1_HOST_MODE_OVERRIDE:** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.<br><br>**Power Well:** Core |
| 17:16 | 0x0<br>RW | **eUSB2SEL:** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2 0: Port is mapped to USB2 1: Port is mapped to eUSB2<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ignore IN EP s. 0 HS ASYNC Active will include IN EP s. 1 HS ASYNC Active will mask/ignore IN EP s.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW | **HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK. 0 HS OUT ASYNC Active will include EP s that are polling. 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.<br><br>**Power Well:** Core |
| 13 | 0x1<br>RW | **HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK. 0 HS IN ASYNC Active will include EP s that are polling. 1 HS IN ASYNC Active will mask/ignore EP s that are polling.<br><br>**Power Well:** Core |
| 12:11 | 0x0<br>RW | **HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active. This will determine how the smallest service interval among active EP s and number of active EP s are used. 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement. 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement. 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication. 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indicaiton<br><br>**Power Well:** Core |
| 10:4 | 0x0<br>RW | **HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Acive may include HS/FS/LS INTR IN EP active indication. If there are more than NumEPThreshold active HS/FS/LS INTR EP s then they may be included as part of the periodic active generation.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Acive will include HS/FS/LS INTR IN EP active indication. If there are any active HS/FS/LS INTR EP s with a service interval less than or equal to this threshold then they may be included as part of the periodic active generation.<br><br>**Power Well:** Core |

## 51.3.238 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h

The RW/L property of this register is controlled by OCCFDONE bit.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 86B0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Rsvd:** Reserved<br><br>**Power Well:** SUS |
| 3:0 | 0x0<br>RW/L | **OC Mapping (OCM):**<br>• 0 - port 0<br>• 1 - port 1<br>• 2 - port 2<br>• 3 - port 3<br>• Others: Reserved<br><br>**Power Well:** SUS |

## 51.3.239 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h

The RW/L property of this register is controlled by OCCFDONE bit.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 86D0h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Rsvd:** Reserved.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW/L | **OC Mapping (OCM):**<br>• 0 - port 0<br>• 1 - port 1<br>• 2 - port 2<br>• 3 - port 3<br>• Others: Reserved<br><br>**Power Well:** SUS |

## 51.3.240 XHCC3 (XHCC3_MMIO)—Offset 86FCh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000002h

**BAR and Offset:** [MBAR] + 86FCh

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (Rsvd1):** Reserved<br><br>**Power Well:** SUS |
| 3 | 0x0<br>RW | **Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface<br><br>**Power Well:** SUS |
| 2 | 0x0<br>RW | **Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface<br><br>**Power Well:** SUS |
| 1 | 0x1<br>RW | **Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected<br><br>**Power Well:** SUS |
| 0 | 0x0<br>RW/L | **Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.<br><br>**Power Well:** SUS |

## 51.3.241 Debug Capability ID Register (DCID)—Offset 8700h

This register is modified and maintained by BIOS

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 0005000Ah

**BAR and Offset:** [MBAR] + 8700h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 20:16 | 0x5<br>RW | **Debug Capability Event Ring Segment Table Max (DCERSTM):** Note: This register is sticky.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RW | **Next Capability Pointer (NCP):** Note: This register is sticky.<br><br>**Power Well:** Core |
| 7:0 | 0xa<br>RW | **Capability ID (CID):** Note: This register is sticky.<br><br>**Power Well:** Core |

## 51.3.242 Debug Capability Doorbell Register (DCDB)—Offset 8704h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8704h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RW | **Doorbell Target (DBTGT):** This field defines the target of the doorbell reference. The table below defines the Debug Capability notification that is generated by ringing the doorbell. Value Definition 0 Data EP 1 OUT Enqueue Pointer Update 1 Data EP 1 IN Enqueue Pointer Update 2:255 Reserved This field returns '0' when read and the value should be treated as undefined by software.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |

## 51.3.243 Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8708h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Event Ring Segment Table Size (ERSTS):** This field identifies the number of valid Event Ring Segment Table entries in the Event Ring Segment Table pointed to by the Debug Capability Event Ring Segment Table Base Address register. The maximum value supported by an xHC implementation for this register is defined by the DCERST Max field in the DCID register Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.<br><br>**Power Well:** Core |

## 51.3.244 Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MBAR] + 8710h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:4 | 0x0<br>RW | **Event Ring Segment Table Base Address Register (ERSTBAR):** This field defines the high order bits of the start address of the Debug Capability Event Ring Segment Table. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 51.3.245 Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MBAR] + 8718h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:4 | 0x0<br>RW | **Dequeue Pointer (DQP):** This field defines the high order bits of the 64-bit address of the current Debug Capability Event Ring Dequeue Pointer. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Dequeue ERST Segment Index (DESI):** This field may be used by the xHC to accelerate checking the Event Ring full condition. This field is written with the low order 3 bits of the offset of the ERST entry which defines the Event Ring segment that the Event Ring Dequeue Pointer resides in.<br><br>**Power Well:** Core |

## 51.3.246 Debug Capability Control Register (DCCTRL)—Offset 8720h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8720h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Debug Capability Enable (DCE):** Reserved.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RO | **Device Address (DADDR):** Reserved.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Debug Max Burst Size (DMBS):** LPT-LP USB Debug Device does not support bursting.<br><br>**Power Well:** Core |
| 15:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/C | **DbC Run Change (DRC):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/1S | **Halt IN TR (HIT):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/1S | **Halt OUT TR (HOT):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Link Status Event Enable (LSE):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **DbC Run (DCR):** Reserved.<br><br>**Power Well:** Core |

## 51.3.247 Debug Capability Status Register (DCST)—Offset 8724h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8724h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Debug Port Number (DPNUM):** This field provides the ID of the Root Hub port that the Debug Capability has been automatically attached to. The value is '0' when the Debug Capability is not attached to a Root Hub port.<br><br>**Power Well:** Core |
| 23:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Event Ring Not Empty (ERNE):** When '1', this field indicates that the Debug Capability Event Ring has a Transfer Event on it. It is automatically cleared to '0' by the xHC when the Debug Capability Event Ring is empty, i.e. the Debug Capability Enqueue Pointer is equal to the Debug Capability Event Ring Dequeue Pointer register.<br><br>**Power Well:** Core |

## 51.3.248 Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000080h

**BAR and Offset:** [MBAR] + 8728h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23 | 0x0<br>RW/C | **Port Config Error Change (CEC):** This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it.<br><br>**Power Well:** Core |
| 22 | 0x0<br>RW/C | **Port Link Status Change (PLC):** This flag is set to '1' due to the following PLS transitions: U0 -) U3 Suspend signaling detected from Debug Host U3 -) U0 Resume complete Polling -) Disabled Training Error Ux or Recovery -) Inactive Error Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'<br><br>**Power Well:** Core |
| 21 | 0x0<br>RW/C | **Port Reset Change (PRC):** This bit is set when reset processing on this port is complete (i.e. a '1' to '0' transition of PR). '0' = No change. '1' = Reset complete.Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.<br><br>**Power Well:** Core |
| 20:18 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW/C | **Connect Status Change (CSC):** an already-set bit (i.e., the bit will remain '1'). Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.<br><br>**Power Well:** Core |
| 16:14 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved.<br><br>**Power Well:** Core |
| 13:10 | 0x0<br>RO | **Port Speed (PSPD):** This field identifies the speed of the port. This field is only relevant when a Debug Host is attached (CCS = '1') in all other cases this field shall indicate Undefined Speed. 0 Undefined Speed 1-15 Protocol Speed ID (PSI) Note: The Debug Capability onlydoes not supports LS, FS, or HS operation.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved.<br><br>**Power Well:** Core |
| 8:5 | 0x4<br>RO | **Port Link State (PLS):** This field reflects its current link state. This field is only relevant when a Debug Host is attached (Debug Port Number ) '0'). Value Meaning 0 Link is in the U0 State 1 Link is in the U1 State 2 Link is in the U2 State 3 Link is in the U3 State (Device Suspended) 4 Link is in the Disabled State 5 Link is in the RxDetect State 6 Link is in the Inactive State 7 Link is in the Polling State 8 Link is in the Recovery State 9 Link is in the Hot Reset State 15:10 Reserved Note: Transitions between different states are not reflected until the transition is complete.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RO | **Port Reset (PR):** '1' = Port is in Reset. '0' = Port is not in Reset. This bit is set to '1' when the bus reset sequence as defined in the USB Specification is detected on the Root Hub port assigned to the Debug capability. It is cleared when the bus reset sequence is completed by the Debug Host, and the DbC shall transition to the USB Default state. A '0' to '1' transition of this bit shall clear DCPORTSC PED ('0'). This field is '0' if DCE or CCS are '0'.<br><br>**Power Well:** Core |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Port Enabled/Disabled (PED):** Default = '0'. '1' = Enabled. '0' = Disabled. This flag shall be set to '1' by a '0' to '1' transition of CCS or a '1' to '0' transition of the PR. When PED transitions from '1' to '0' due to the assertion of PR, the port's link shall transition to the Rx.Detect state. This flag may be used by software to enable or disable the operation of the Root Hub port assigned to the Debug Capability. The Debug Capability Root Hub port operation may be disabled by a fault condition (disconnect event or other fault condition, e.g. a LTSSM Polling substate timeout, tPortConfiguration timeout error, etc.), the assertion of DCPORTSC PR, or by software. 0 = Debug Capability Root Hub port is disabled. 1 = Debug Capability Root Hub port is enabled. When the port is disabled (PED = '0') the port's link shall enter the SS.Disabled state and remain there until PED is reasserted ('1') or DCE is negated ('0'). Note that the Root Hub port is remains mapped to Debug Capability while PED = '0'. While PED = '0' the Debug Capability will appear to be disconnected to the Debug Host. Note, this bit is not affected by PORTSC PR bit transitions. This field is '0' if DCE or CCS are '0'.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Current Connect Status (CCS):** '1' = A Root Hub port is connected to a Debug Host and assigned to the Debug Capability. '0' = No Debug Host is present. This value reflects the current state of the port, and may not correspond to the value reported by the Connect Status Change (CSC) field in the Port Status Change Event that was generated by a '0' to '1' transition of this bit. This flag is '0' if Debug Capability Enable (DCE) is '0'.<br><br>**Power Well:** Core |

## 51.3.249 Debug Capability Context Pointer Register (DCCP)—Offset 8730h

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [MBAR] + 8730h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:4 | 0x0<br>RW | **Debug Capability Context Pointer Register (DCCPR):** This field defines the high order bits of the start address of the Debug Capability Context data structure associated with the Debug Capability. Software shall initialize this register before setting the Debug Capability Enable bit in the Debug Capability Control Register to '1'.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 51.3.250 Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 80870000h

**BAR and Offset:** [MBAR] + 8738h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x8087<br>RW | **Vendor ID (VID):** This field is presented by the Debug Device in the USB Device Descriptor idVendor field.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **DbC Protocol (DBCPR):** This field is presented by the Debug Device in the USB Interface Descriptor bInterfaceProtocol field. Value Function 0 Debug Target vendor defined. 1 GNU Remote Debug Command Set supported. 2-255 Reserved.<br><br>**Power Well:** Core |

## 51.3.251 Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 873Ch

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Device Revision (DREV):** This field is presented by the Debug Device in the USB Device Descriptor bcdDevice field.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Product ID (PID):** This field is presented by the Debug Device in the USB Device Descriptor idProduct field.<br><br>**Power Well:** Core |

## 51.3.252 Debug Capability Descriptor Parameters (DCDP)—Offset 8740h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000030C3h

**BAR and Offset:** [MBAR] + 8740h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RW | **Max Power Field (MPF):** This field will be used by USB Debug Device to report maximum power consumption when the device is fully operational. This value is returned by bMaxPower field in response to Configuration Descriptor read from the debug device. Note: bU1DevExitLat and bU2DevExitLat fields returned in BOS Descriptor read will be taken from the corresponding fields from the Host Controller space.<br><br>**Power Well:** Core |
| 15:8 | 0x30<br>RW/S | **NEXT CAPABILITY POINTER (NCP):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0xc3<br>RW/S | **Capability ID (CID):** Reserved.<br><br>**Power Well:** Core |

## 51.3.253 Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h

This register contains a number of fields that provide a specific level of configurability for the OUT DMA that is part of Debug Device logic. This configurability is above and beyond that defined in the xHCI specification.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MBAR] + 8748h

**Default:** 00000000h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RW | **Enable ACK FIFO credit accounting (EN_ACK_FCA):** Setting this field will enable ACK FIFO credit accounting. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP<br><br>**Power Well:** Core |
| 17:14 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Enable ACK FIFO ICA mechanisms (EN_ACK_FIFO_ICA):** Setting this field will enable ACK FIFO individual credit accounting mechanisms for Async vs. Periodic Endpoints. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP<br><br>**Power Well:** Core |
| 12:9 | 0x0<br>RW | **Reserved (RSVD_2):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Clear ownership of context semaphore (CL_OWN_CS):** Setting this field generates a pulse that clears the ownership of the context semaphore that is shared between the Out DMA Response and Completion Finite State Machines<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **Return OD ACK credits (RET_OD_ACK_CR):** Setting this field generates a pulse that implicitly returns all of the Out DMA ACK credits on all ports<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Reserved (RSVD_3):** Reserved.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **Return ODCF SM to idle state (RET_ODCF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Completion Finite State Machine into the IDLE state<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW | **Return ODRF SM to idle state (RET_ODRF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Response Finite State Machine into the IDLE state<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **Return ODRDF SM to idle state (RET_ODRDF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Read Finite State Machine into the IDLE state<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2:0 | 0x0<br>RW | **Reserved (RSVD_4):** Reserved.<br><br>**Power Well:** Core |

## 51.3.254 DBC Control Register 1 (DBCCTL_REG)—Offset 8760h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MBAR] + 8760h

**MBAR Type:** PCI Configuration Register (Size: 64 bits)
**MBAR Reference:** [B:0, D:21, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** RSVD<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **SW_DCE_SEL:** Reserved.<br><br>**Power Well:** Core |
| 6:3 | 0x0<br>RW | **DISC_RXD_CNT:** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **RSVD0:** Reserved |
| 1 | 0x0<br>RW | **Force DCE Mode (FORCE_DCE_MODE):** 0: When DCE is set, the DbC switches to Mode 2 1: When DCE is set, the DbC switches to Mode 3<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Force Disconnect upon DCE (FORCE_DISCONNECT_ON_DCE):** If this bit is set by BIOS, the DbC will temporarily disconnect from the remote host if the DCE is set, and shortly thereafter re-connect. This allows the DbC to switch from Mode1 to Mode2 or Mode 3 operation upon DCE being set.<br><br>**Power Well:** Core |

§ §

# 52 System Management - B0, D24, F(0, 1, 3, 4)

## 52.1 Introduction and Index

The host-accessible registers for the System Management are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 52.1.1 Host Configuration Space—HECI1 (F:0)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 24 (decimal), Function 0. The offset addresses are listed.

**Table 52-1. Summary of PCI Configuration Registers—0/24/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19D38086 | "Identifiers (HECI1_ID)—Offset 0h" on page 2239 |
| 4 | 2 | 0000 | "Command (HECI1_CMD)—Offset 4h" on page 2240 |
| 6 | 2 | 0010 | "Status (HECI1_STS)—Offset 6h" on page 2241 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI1_RID_CC)—Offset 8h" on page 2242 |
| C | 1 | 00 | "Cache Line Size (HECI1_CLS)—Offset Ch" on page 2242 |
| D | 1 | 00 | "Master Latency Timer (HECI1_MLT)—Offset Dh" on page 2242 |
| E | 1 | 80 | "Header Type (HECI1_HTYPE)—Offset Eh" on page 2243 |
| F | 1 | 00 | "Built In Self-Test (HECI1_BIST)—Offset Fh" on page 2243 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h" on page 2243 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h" on page 2244 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI1_SS)—Offset 2Ch" on page 2244 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI1_CAP)—Offset 34h" on page 2244 |
| 3C | 2 | 0100 | "Interrupt Information (HECI1_INTR)—Offset 3Ch" on page 2245 |
| 3E | 1 | 00 | "Minimum Grant (HECI1_MGNT)—Offset 3Eh" on page 2245 |
| 3F | 1 | 00 | "Maximum Latency (HECI1_MLAT)—Offset 3Fh" on page 2245 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI1_HFS)—Offset 40h" on page 2246 |
| 44 | 4 | 80000000 | "Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h" on page 2248 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h" on page 2249 |
| 4C | 4 | 00000000 | "Host General Status (HECI1_H_GS1)—Offset 4Ch" on page 2250 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI1_PID)—Offset 50h" on page 2250 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI1_PC)—Offset 52h" on page 2250 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h" on page 2251 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h" on page 2251 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h" on page 2252 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h" on page 2252 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch" on page 2252 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI1_H_GS2)—Offset 70h" on page 2252 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI1_H_GS3)—Offset 74h" on page 2253 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch" on page 2253 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh" on page 2253 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h" on page 2254 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h" on page 2254 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h" on page 2254 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h" on page 2255 |
| BC | 4 | 40000000 | "Host Extend Register Status (HECI1_HERS)—Offset BCh" on page 2255 |
| C0 | 4 | 00000000 | "Host Extend Register DW1. (HECI1_HER1)—Offset C0h" on page 2256 |

## Table 52-1. Summary of PCI Configuration Registers—0/24/0 (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| C4 | 4 | 00000000 | "Host Extend Register DW2. (HECI1_HER2)—Offset C4h" on page 2256 |
| C8 | 4 | 00000000 | "Host Extend Register DW3. (HECI1_HER3)—Offset C8h" on page 2256 |
| CC | 4 | 00000000 | "Host Extend Register DW4. (HECI1_HER4)—Offset CCh" on page 2256 |
| D0 | 4 | 00000000 | "Host Extend Register DW5. (HECI1_HER5)—Offset D0h" on page 2257 |
| D4 | 4 | 00000000 | "Host Extend Register DW6. (HECI1_HER6)—Offset D4h" on page 2257 |
| D8 | 4 | 00000000 | "Host Extend Register DW7. (HECI1_HER7)—Offset D8h" on page 2257 |
| DC | 4 | 00000000 | "Host Extend Register DW8. (HECI1_HER8)—Offset DCh" on page 2257 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI1_MANID)—Offset F8h" on page 2258 |

### 52.1.2 Host Configuration Space—HECI2 (F:1)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 24 (decimal), Function 1. The offset addresses are listed.

**Table 52-2. Summary of PCI Configuration Registers—0/24/1**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19D48086 | "Identifiers (HECI2_ID)—Offset 0h" on page 2259 |
| 4 | 2 | 0000 | "Command (HECI2_CMD)—Offset 4h" on page 2259 |
| 6 | 2 | 0010 | "Status (HECI2_STS)—Offset 6h" on page 2261 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI2_RID_CC)—Offset 8h" on page 2262 |
| C | 1 | 00 | "Cache Line Size (HECI2_CLS)—Offset Ch" on page 2262 |
| D | 1 | 00 | "Master Latency Timer (HECI2_MLT)—Offset Dh" on page 2262 |
| E | 1 | 80 | "Header Type (HECI2_HTYPE)—Offset Eh" on page 2263 |
| F | 1 | 00 | "Built In Self-Test (HECI2_BIST)—Offset Fh" on page 2263 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h" on page 2263 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h" on page 2264 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI2_SS)—Offset 2Ch" on page 2264 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI2_CAP)—Offset 34h" on page 2264 |
| 3C | 2 | 0100 | "Interrupt Information (HECI2_INTR)—Offset 3Ch" on page 2265 |
| 3E | 1 | 00 | "Minimum Grant (HECI2_MGNT)—Offset 3Eh" on page 2265 |
| 3F | 1 | 00 | "Maximum Latency (HECI2_MLAT)—Offset 3Fh" on page 2265 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI2_HFS)—Offset 40h" on page 2266 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h" on page 2268 |
| 4C | 4 | 00000000 | "Host General Status (HECI2_H_GS1)—Offset 4Ch" on page 2269 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI2_PID)—Offset 50h" on page 2269 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI2_PC)—Offset 52h" on page 2270 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h" on page 2271 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h" on page 2271 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h" on page 2272 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h" on page 2272 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch" on page 2272 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI2_H_GS2)—Offset 70h" on page 2272 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI2_H_GS3)—Offset 74h" on page 2273 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch" on page 2273 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh" on page 2273 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h" on page 2274 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h" on page 2274 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h" on page 2274 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h" on page 2275 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI2_MANID)—Offset F8h" on page 2275 |

## 52.1.3 Host Configuration Space—HECI3 (F:4)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 24 (decimal), Function 4. The offset addresses are listed.

**Table 52-3. Summary of PCI Configuration Registers—0/24/4**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19D68086 | "Identifiers (HECI3_ID)—Offset 0h" on page 2276 |
| 4 | 2 | 0000 | "Command (HECI3_CMD)—Offset 4h" on page 2277 |
| 6 | 2 | 0010 | "Status (HECI3_STS)—Offset 6h" on page 2278 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI3_RID_CC)—Offset 8h" on page 2279 |
| C | 1 | 00 | "Cache Line Size (HECI3_CLS)—Offset Ch" on page 2279 |
| D | 1 | 00 | "Master Latency Timer (HECI3_MLT)—Offset Dh" on page 2279 |
| E | 1 | 80 | "Header Type (HECI3_HTYPE)—Offset Eh" on page 2280 |
| F | 1 | 00 | "Built In Self-Test (HECI3_BIST)—Offset Fh" on page 2280 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h" on page 2280 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h" on page 2281 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI3_SS)—Offset 2Ch" on page 2281 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI3_CAP)—Offset 34h" on page 2281 |
| 3C | 2 | 0100 | "Interrupt Information (HECI3_INTR)—Offset 3Ch" on page 2282 |
| 3E | 1 | 00 | "Minimum Grant (HECI3_MGNT)—Offset 3Eh" on page 2282 |
| 3F | 1 | 00 | "Maximum Latency (HECI3_MLAT)—Offset 3Fh" on page 2282 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI3_HFS)—Offset 40h" on page 2283 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h" on page 2285 |
| 4C | 4 | 00000000 | "Host General Status (HECI3_H_GS1)—Offset 4Ch" on page 2286 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI3_PID)—Offset 50h" on page 2286 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI3_PC)—Offset 52h" on page 2286 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h" on page 2287 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h" on page 2287 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h" on page 2288 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h" on page 2288 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch" on page 2288 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI3_H_GS2)—Offset 70h" on page 2288 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI3_H_GS3)—Offset 74h" on page 2289 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch" on page 2289 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh" on page 2289 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h" on page 2290 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h" on page 2290 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h" on page 2290 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h" on page 2291 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI3_MANID)—Offset F8h" on page 2291 |

### 52.1.4 Host Configuration Space—KT (F:3)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 24 (decimal), Function 3. The offset addresses are listed.

**Table 52-4. Summary of PCI Configuration Registers—0/24/3**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19D58086 | "Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h" on page 2292 |
| 4 | 4 | 00B00000 | "Status and Command (KT_HOST_STS_CMD)—Offset 4h" on page 2293 |
| 8 | 4 | 07000200 | "Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h" on page 2295 |
| C | 4 | 00800000 | "BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch" on page 2295 |
| 10 | 4 | 00000001 | "KT IO BAR (KT_HOST_IOBAR)—Offset 10h" on page 2296 |
| 14 | 4 | 00000000 | "KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h" on page 2296 |
| 28 | 4 | 00000000 | "Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h" on page 2297 |
| 2C | 4 | 00008086 | "Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch" on page 2297 |
| 30 | 4 | 00000000 | "Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h" on page 2297 |
| 34 | 4 | 00000040 | "Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h" on page 2298 |
| 3C | 4 | 00000000 | "Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch" on page 2298 |
| 40 | 4 | 00805005 | "MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h" on page 2299 |
| 44 | 4 | 00000000 | "MSI Message Address (KT_HOST_MSIMA)—Offset 44h" on page 2299 |
| 48 | 4 | 00000000 | "MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h" on page 2300 |
| 4C | 4 | 00000000 | "MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch" on page 2300 |
| 50 | 4 | 00230001 | "Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h" on page 2301 |
| 54 | 4 | 00000008 | "Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h" on page 2302 |
| F8 | 4 | 00000000 | "Process Stepping Information (KT_HOST_PSI)—Offset F8h" on page 2303 |

## 52.1.5 Host Configuration Space—IDE (F:2)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 24 (decimal), Function 2. The offset addresses are listed.

**Table 52-5. Summary of PCI Configuration Registers—0/24/2**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19EA8086 | "Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h" on page 2304 |
| 4 | 4 | 00B00000 | "Status and Command (IDE_HOST_STS_CMD)—Offset 4h" on page 2305 |
| 8 | 4 | 01018500 | "Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h" on page 2307 |
| C | 4 | 00800000 | "BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch" on page 2307 |
| 10 | 4 | 00000001 | "IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h" on page 2308 |
| 14 | 4 | 00000001 | "IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h" on page 2308 |
| 18 | 4 | 00000001 | "IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h" on page 2309 |
| 1C | 4 | 00000001 | "IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch" on page 2309 |
| 20 | 4 | 00000001 | "IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h" on page 2310 |
| 28 | 4 | 00000000 | "Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h" on page 2310 |
| 2C | 4 | 00008086 | "Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch" on page 2311 |
| 30 | 4 | 00000000 | "Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h" on page 2311 |
| 34 | 4 | 00000040 | "Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h" on page 2311 |
| 3C | 4 | 00000000 | "Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch" on page 2312 |
| 40 | 4 | 00805005 | "MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h" on page 2313 |
| 44 | 4 | 00000000 | "MSI Message Address (IDE_HOST_MSIMA)—Offset 44h" on page 2313 |
| 48 | 4 | 00000000 | "MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h" on page 2314 |
| 4C | 4 | 00000000 | "MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch" on page 2314 |
| 50 | 4 | 00230001 | "Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h" on page 2315 |
| 54 | 4 | 00000008 | "Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h" on page 2316 |
| F8 | 4 | 00000000 | "Process Stepping Information (IDE_HOST_PSI)—Offset F8h" on page 2317 |

### 52.1.6 Host Memory Space—HECI1_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 52-6. Summary of Memory Mapped I/O Registers—HECI1_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI1_H_CB_WW)—Offset 0h" on page 2318 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI1_H_CSR)—Offset 4h" on page 2319 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h" on page 2320 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch" on page 2320 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI1_D0I3C)—Offset 800h" on page 2321 |

### 52.1.7 Host Memory Space—HECI2_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 52-7. Summary of Memory Mapped I/O Registers—HECI2_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI2_H_CB_WW)—Offset 0h" on page 2322 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI2_H_CSR)—Offset 4h" on page 2323 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h" on page 2324 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch" on page 2324 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI2_D0I3C)—Offset 800h" on page 2325 |

### 52.1.8 Host Memory Space—HECI3_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 52-8. Summary of Memory Mapped I/O Registers—HECI3_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI3_H_CB_WW)—Offset 0h" on page 2326 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI3_H_CSR)—Offset 4h" on page 2327 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h" on page 2328 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch" on page 2328 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI3_D0I3C)—Offset 800h" on page 2329 |

### 52.1.9 Host Memory Space—KT_HOST_MEMBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 52-9. Summary of Memory Mapped I/O Registers—KT_HOST_MEMBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "KT Receive Buffer Register (MEM_KTRBR)—Offset 0h" on page 2330 |
| 0 | 1 | 00 | "KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h" on page 2330 |
| 0 | 1 | 00 | "KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h" on page 2331 |
| 1 | 1 | 00 | "KT Interrupt Enable register (MEM_KTIER)—Offset 1h" on page 2331 |
| 1 | 1 | 00 | "KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h" on page 2332 |
| 2 | 1 | 01 | "KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h" on page 2333 |
| 2 | 1 | 00 | "KT FIFO Control register (MEM_KTFCR)—Offset 2h" on page 2334 |
| 3 | 1 | 03 | "KT Line Control register (MEM_KTLCR)—Offset 3h" on page 2335 |
| 4 | 1 | 00 | "KT Modem Control register (MEM_KTMCR)—Offset 4h" on page 2336 |
| 5 | 1 | 60 | "KT Line Status register (MEM_KTLSR)—Offset 5h" on page 2337 |
| 6 | 1 | 00 | "KT Modem Status register (MEM_KTMSR)—Offset 6h" on page 2338 |
| 7 | 1 | 00 | "KT Scratch register (MEM_KTSCR)—Offset 7h" on page 2338 |

## 52.1.10 Host Memory Space—Fixed Address

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 52-10. Summary of Memory Mapped I/O Registers—Fixed Addresses**

| Memory Address (hex) | Bytes (decimal) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| FED40000 | 4 | 00000000 | "FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h" on page 2339 |
| FED40004 | 4 | 00000000 | "FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h" on page 2340 |
| FED40008 | 4 | 00000000 | "FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h" on page 2340 |
| FED4000C | 4 | 00000000 | "FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch" on page 2340 |
| FED40010 | 4 | 00000000 | "FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h" on page 2341 |
| FED40030 | 4 | 00000000 | "FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h" on page 2341 |
| FED40034 | 4 | 00000000 | "FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h" on page 2341 |
| FED40038 | 4 | 00000000 | "FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h" on page 2342 |
| FED40040 | 4 | 00000000 | "Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h" on page 2342 |
| FED40044 | 4 | 00000000 | "Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h" on page 2342 |
| FED40048 | 4 | 00000000 | "Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h" on page 2343 |
| FED4004C | 4 | 00000000 | "Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch" on page 2343 |
| FED40050 | 4 | 00000000 | "Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h" on page 2343 |
| FED40054 | 4 | 00000000 | "Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h" on page 2344 |
| FED40058 | 4 | 00000000 | "Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h" on page 2344 |
| FED4005C | 4 | 00000000 | "Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch" on page 2344 |
| FED40060 | 4 | 00000000 | "Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h" on page 2345 |
| FED40064 | 4 | 00000000 | "Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h" on page 2345 |
| FED40068 | 4 | 00000000 | "Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h" on page 2345 |
| FED4006C | 4 | 00000000 | "Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch" on page 2346 |
| FED40070 | 4 x 4 | 00000000 | "FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h" on page 2346 |
| FED40080 | 4 x 992 | 00000000 | "FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h" on page 2346 |

## 52.1.11 Host I/O Space—KT_HOST_IOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 52-11. Summary of I/O Registers—KT_HOST_IOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "KT Receive Buffer Register (IO_KTRBR)—Offset 0h" on page 2347 |
| 0 | 1 | 00 | "KT Transmitter Holding Register (IO_KTTHR)—Offset 0h" on page 2347 |
| 0 | 1 | 00 | "KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h" on page 2348 |
| 1 | 1 | 00 | "KT Interrupt Enable register (IO_KTIER)—Offset 1h" on page 2348 |
| 1 | 1 | 00 | "KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h" on page 2349 |
| 2 | 1 | 01 | "KT Interrupt Identification Register (IO_KTIIR)—Offset 2h" on page 2349 |
| 2 | 1 | 00 | "KT FIFO Control register (IO_KTFCR)—Offset 2h" on page 2350 |
| 3 | 1 | 03 | "KT Line Control register (IO_KTLCR)—Offset 3h" on page 2351 |
| 4 | 1 | 00 | "KT Modem Control register (IO_KTMCR)—Offset 4h" on page 2352 |
| 5 | 1 | 60 | "KT Line Status register (IO_KTLSR)—Offset 5h" on page 2353 |
| 6 | 1 | 00 | "KT Modem Status register (IO_KTMSR)—Offset 6h" on page 2354 |
| 7 | 1 | 00 | "KT Scratch register (IO_KTSCR)—Offset 7h" on page 2355 |

### 52.1.12 Host I/O Space—IDE_HOST_PCMDIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 52-12. Summary of I/O Registers—IDE_HOST_PCMDIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 0000 | "IDE Data Register (IDEDATA)—Offset 0h" on page 2356 |
| 1 | 1 | 00 | "IDE Features Register (IDEFR)—Offset 1h" on page 2356 |
| 1 | 1 | 00 | "IDE Error Register DEV0 (IDEERD0)—Offset 1h" on page 2357 |
| 1 | 1 | 00 | "IDE Error Register DEV1 (IDEERD1)—Offset 1h" on page 2357 |
| 2 | 1 | 00 | "IDE Sector Count In Register (IDESCIR)—Offset 2h" on page 2358 |
| 2 | 1 | 00 | "IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h" on page 2358 |
| 2 | 1 | 00 | "IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h" on page 2358 |
| 3 | 1 | 00 | "IDE Sector Number In Register (IDESNIR)—Offset 3h" on page 2359 |
| 3 | 1 | 00 | "IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h" on page 2359 |
| 3 | 1 | 00 | "IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h" on page 2360 |
| 4 | 1 | 00 | "IDE Cylinder Low In Register (IDECLIR)—Offset 4h" on page 2360 |
| 4 | 1 | 00 | "IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h" on page 2361 |
| 4 | 1 | 00 | "IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h" on page 2361 |
| 5 | 1 | 00 | "IDE Cylinder High In Register (IDECHIR)—Offset 5h" on page 2362 |
| 5 | 1 | 00 | "IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h" on page 2362 |
| 5 | 1 | 00 | "IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h" on page 2363 |
| 6 | 1 | 00 | "IDE Drive/Head In Register (IDEDHIR)—Offset 6h" on page 2363 |
| 6 | 1 | 00 | "IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h" on page 2364 |
| 6 | 1 | 00 | "IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h" on page 2364 |
| 7 | 1 | 00 | "IDE Command Register (IDECR)—Offset 7h" on page 2365 |
| 7 | 1 | 80 | "IDE Status Register DEV0 (IDESR0)—Offset 7h" on page 2366 |
| 7 | 1 | 80 | "IDE Status Register DEV1 (IDESR1)—Offset 7h" on page 2367 |

### 52.1.13 Host I/O Space—IDE_HOST_PCTLIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 52-13. Summary of I/O Registers—IDE_HOST_PCTLIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2 | 1 | 00 | "IDE Device Control Register (IDDCR)—Offset 2h" on page 2368 |
| 2 | 1 | 00 | "IDE Alternate Status Register (IDASR)—Offset 2h" on page 2368 |

### 52.1.14 Host I/O Space—IDE_HOST_BMIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 52-14. Summary of I/O Registers—IDE_HOST_BMIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h" on page 2369 |
| 1 | 1 | 00 | "IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h" on page 2370 |
| 2 | 1 | 80 | "IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h" on page 2371 |
| 3 | 1 | 00 | "IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h" on page 2373 |
| 4 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h" on page 2373 |
| 5 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h" on page 2373 |
| 6 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h" on page 2374 |
| 7 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h" on page 2374 |
| 8 | 1 | 00 | "IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h" on page 2375 |
| 9 | 1 | 00 | "IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h" on page 2375 |
| A | 1 | 80 | "IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah" on page 2376 |
| B | 1 | 00 | "IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh" on page 2377 |
| C | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch" on page 2377 |
| D | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh" on page 2378 |
| E | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh" on page 2378 |
| F | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh" on page 2379 |

### 52.1.15 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 52-15. Summary of Sideband Registers for KT (Host)—0xe4**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 0 | 00000000 | "KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h" on page 2380 |
| 4 | 00000000 | "BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h" on page 2380 |

### Table 52-16. Summary of Sideband Registers for IDE (Host)—0xe4

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 0 | 00000000 | "IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h" on page 2381 |
| 4 | 00000000 | "BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h" on page 2381 |

## 52.2 Registers in Configuration Space—HECI1

### 52.2.1 Identifiers (HECI1_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 0h

**Default:** 19D38086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19d3 RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086 RO | **Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG. |

## 52.2.2 Command (HECI1_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operation. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's abiity to act as a system memory master for data transfers.<br>When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU.<br>***Note:*** This bit does not block HECI accesses to CSE-UMA |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controllers memory mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

### 52.2.3 Status (HECI1_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Parity Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>This bit is a 1 when the INTx# is asserted.<br>This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

## 52.2.4 Revision ID and Class Code (HECI1_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

## 52.2.5 Cache Line Size (HECI1_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

## 52.2.6 Master Latency Timer (HECI1_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

## 52.2.7 Header Type (HECI1_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi- function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

## 52.2.8 Built In Self-Test (HECI1_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved. |

## 52.2.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable. |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

## 52.2.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

## 52.2.11 Sub System Identifiers (HECI1_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

## 52.2.12 Capabilities Pointer (HECI1_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 52.2.13 Interrupt Information (HECI1_INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 52.2.14 Minimum Grant (HECI1_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 52.2.15 Maximum Latency (HECI1_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 52.2.16 Host Firmware Status (HECI1_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 52.2.17 Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 44h

**Default:** 80000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Miscellaneous Shadow Valid (MSVLD):** This bit is hardwired to 1 to indicate that this HECI device implements the Miscellaneous Shadow register.<br>This bit can be used by host software that is bus/dev/function number agnostic (such as HECI operating system driver) to discover whether the Miscellaneous Shadow register is implemented or not. |
| 30:17 | 0x0<br>RO | **RSVD_30_17:** Reserved. |
| 16 | 0x0<br>RO/V | **CSE UMA Size Valid (CUSZV):** This bit indicates that FW has written the CUSZ field.<br>This field reflects the value of CSE CUBA.CUSZV. |
| 15:7 | 0x0<br>RO | **Reserved (RSVD_15_7):** Reserved. |
| 6:0 | 0x0<br>RO/V | **CSE UMA Size (CUSZ):** These bits reflect firmware's desired size of CSEUMA memory region.<br>It is configured by firmware prior to bring up core power and allowing BIOS to initialize memory.<br>The legal CSEUMA sizes and encodings for this field are:<br>• 0000000b (0MB, no CSEUMA region)<br>• 0000001b (1MB)<br>• 0000010b (2MB)<br>• 0000100b (4MB)<br>• 0001000b (8MB)<br>• 0010000b (16MB)<br>• 0100000b (32MB)<br>• 1000000b (64MB).<br>This field reflects the value of CSE CUBA.CUSZ. |

## 52.2.18 General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 52.2.19 Host General Status (HECI1_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host.<br>This field is not used by hardware. |

## 52.2.20 PCI Power Management Capability ID (HECI1_PID)—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list.<br>This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 52.2.21 PCI Power Management Capabilities (HECI1_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 52.2.22 PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it.<br>CSE cannot clear this bit.<br>Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit.<br>When reset these messages may not be sent. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state.<br>The D1 and D2 states are not supported for this HECI host controller.<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.<br>When in the D3HOT state, the HBA's configuration space is available, but the registers memory space is not.<br>Additionally, interrupts are blocked.<br>This field is visible to firmware through the H_PCI_CSR register, and changes to this field may be configured by the H_PCI_CSR register to generate an MSI to mIA. |

## 52.2.23 General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 52.2.24 General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 52.2.25 General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 52.2.26 General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 52.2.27 Host General Status 2 (HECI1_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host.<br>This field is not used by hardware. |

## 52.2.28 Host General Status 3 (HECI1_H_GS3)—Offset 74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 52.2.29 Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be other capability pointers (such as PCI-X or PCI- Express) or it can be the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates MSI. |

## 52.2.30 Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

### 52.2.31 Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

### 52.2.32 Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

### 52.2.33 Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:0] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 52.2.34 HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:0] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt;<br>• 01 - Generate SCI;<br>• 10 - Generate SMI;<br>This field may be locked by writing 1 to HIDM_L bit. |

## 52.2.35 Host Extend Register Status (HECI1_HERS)—Offset BCh

This register is used to communicate the CSE FW measurement status information to host.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + BCh

**Default:** 40000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Extend Register Valid (ERV):** Set by FW after all FW has been loaded and the measurement data has been stored in HERx registers.<br>If ERA field is SHA-1, the result of the extend operation is in HER:5-1.<br>If ERA field is SHA-256, the result is in HER:8-1. |
| 30 | 0x1<br>RO | **Extend Feature Present (EFP):** This bit is hardwired to 1 to allow driver software to easily detect the chipset supports the Extend Registers FW measurement feature. |
| 29:4 | 0x0<br>RO | **Reserved (RSVD_29_4):** Reserved. |
| 3:0 | 0x0<br>RO/V | **Extend Register Algorithm (ERA):** This field indicates the hash algorithm used in the FW measurement Extend operation.<br>Encodings are:<br>0x0: SHA-1,<br>0x2: SHA-256.<br>Other values: reserved.<br>This field is set by FW with the used hash algorithm value when the ERV bit is set to 1.<br>This field is meaningless when the ERV bit is 0.<br>This field does NOT have any defined reset value. |

## 52.2.36 Host Extend Register DW1. (HECI1_HER1)—Offset C0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW1 (ERDW1):** Reserved. |

## 52.2.37 Host Extend Register DW2. (HECI1_HER2)—Offset C4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW2 (ERDW2):** Reserved. |

## 52.2.38 Host Extend Register DW3. (HECI1_HER3)—Offset C8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW3 (ERDW3):** Reserved. |

## 52.2.39 Host Extend Register DW4. (HECI1_HER4)—Offset CCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW4 (ERDW4):** Reserved. |

## 52.2.40 Host Extend Register DW5. (HECI1_HER5)—Offset D0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW5 (ERDW5):** Reserved. |

## 52.2.41 Host Extend Register DW6. (HECI1_HER6)—Offset D4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW6 (ERDW6):** Reserved. |

## 52.2.42 Host Extend Register DW7. (HECI1_HER7)—Offset D8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW7 (ERDW7):** Reserved. |

## 52.2.43 Host Extend Register DW8. (HECI1_HER8)—Offset DCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + DCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW8 (ERDW8):** Reserved. |

## 52.2.44 Manufacturer's ID (HECI1_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:0] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** Set by Intel. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process. The dot portion of the process is reflected in bits [27:24]. Note that this field can be used by software to differentiate steppings when Revision ID may not change. |

## 52.3 Registers in Configuration Space—HECI2

**AR Author and Tech Comm:** HIDE this sub-chapter if the device has no such registers.

### 52.3.1 Identifiers (HECI2_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 0h

**Default:** 19D48086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19d4<br>RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG. |

### 52.3.2 Command (HECI2_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operation. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's ability to act as a system memory master for data transfers. When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU. Note that this bit does not block HECI accesses to CSE-UMA |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controllers memory mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

### 52.3.3 Status (HECI2_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>This bit is a 1 when the INTx# is asserted.<br>This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

## 52.3.4 Revision ID and Class Code (HECI2_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

## 52.3.5 Cache Line Size (HECI2_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

## 52.3.6 Master Latency Timer (HECI2_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

## 52.3.7 Header Type (HECI2_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi- function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

## 52.3.8 Built In Self-Test (HECI2_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved |

## 52.3.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

## 52.3.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

## 52.3.11 Sub System Identifiers (HECI2_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

## 52.3.12 Capabilities Pointer (HECI2_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 52.3.13 Interrupt Information (HECI2_INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 52.3.14 Minimum Grant (HECI2_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 52.3.15 Maximum Latency (HECI2_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 52.3.16 Host Firmware Status (HECI2_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 52.3.17 General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 52.3.18 Host General Status (HECI2_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host. This field is not used by hardware. |

## 52.3.19 PCI Power Management Capability ID (HECI2_PID)—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list. This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 52.3.20 PCI Power Management Capabilities (HECI2_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 52.3.21 PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it. CSE cannot clear this bit.<br>Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit.<br>When reset these messages may not be sent. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state.<br>The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state.<br>The D1 and D2 states are not supported for this HECI host controller. |

## 52.3.22 General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 52.3.23 General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 52.3.24 General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 52.3.25 General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 52.3.26 Host General Status 2 (HECI2_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host.<br>This field is not used by hardware. |

## 52.3.27 Host General Status 3 (HECI2_H_GS3)—Offset 74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 52.3.28 Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be other capability pointers (such as PCI-X or PCI- Express) or it can be the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates MSI. |

## 52.3.29 Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

## 52.3.30 Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 52.3.31 Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

## 52.3.32 Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:1] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 52.3.33 HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:1] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt;<br>• 01 - Generate SCI;<br>• 10 - Generate SMI;<br>This field may be locked by writing 1 to HIDM_L bit. |

## 52.3.34 Manufacturer's ID (HECI2_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:1] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part.<br>***Note:*** This field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** This field is set by Intel. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process.<br>The dot portion of the process is reflected in bits [27:24].<br>***Note:*** This field can be used by software to differentiate steppings when Revision ID may not change. |

## 52.4 Registers in Configuration Space—HECI3

### 52.4.1 Identifiers (HECI3_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 0h

**Default:** 19D68086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19d6<br>RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG. |

## 52.4.2 Command (HECI3_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operation. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's ability to act as a system memory master for data transfers.<br>When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU.<br>***Note:*** This bit does not block HECI accesses to CSE-UMA |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controllers memory mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

## 52.4.3 Status (HECI3_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Parity Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>This bit is a 1 when the INTx# is asserted.<br>This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

## 52.4.4 Revision ID and Class Code (HECI3_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

## 52.4.5 Cache Line Size (HECI3_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

## 52.4.6 Master Latency Timer (HECI3_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

## 52.4.7 Header Type (HECI3_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi- function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

## 52.4.8 Built In Self-Test (HECI3_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved |

## 52.4.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

## 52.4.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

## 52.4.11 Sub System Identifiers (HECI3_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

## 52.4.12 Capabilities Pointer (HECI3_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 52.4.13 Interrupt Information (HECI3_INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 52.4.14 Minimum Grant (HECI3_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 52.4.15 Maximum Latency (HECI3_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 52.4.16 Host Firmware Status (HECI3_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 52.4.17 General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 52.4.18 Host General Status (HECI3_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host.<br>This field is not used by hardware. |

## 52.4.19 PCI Power Management Capability ID (HECI3_PID)—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list.<br>This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 52.4.20 PCI Power Management Capabilities (HECI3_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 52.4.21 PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it.<br>CSE cannot clear this bit.<br>Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit.<br>When reset these messages may not be sent. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state.<br>The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state.<br>The D1 and D2 states are not supported for this HECI host controller. |

## 52.4.22 General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 52.4.23 General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 52.4.24 General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 52.4.25 General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 52.4.26 Host General Status 2 (HECI3_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware. |

## 52.4.27 Host General Status 3 (HECI3_H_GS3)—Offset 74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 52.4.28 Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be other capability pointers (such as PCI-X or PCI- Express) or it can be the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates MSI. |

## 52.4.29 Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

## 52.4.30 Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 52.4.31 Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

## 52.4.32 Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:24, F:4] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 52.4.33 HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:24, F:4] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt;<br>• 01 - Generate SCI;<br>• 10 - Generate SMI;<br>This field may be locked by writing 1 to HIDM_L bit. |

## 52.4.34 Manufacturer's ID (HECI3_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:4] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part.<br>***Note:*** Note that this field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** This field is set by Intel. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process.<br>The dot portion of the process is reflected in bits [27:24].<br>***Note:*** This field can be used by software to differentiate steppings when Revision ID may not change.<br>This field is set by a special IOSF SB message (SetIDValue). |

# 52.5 Registers in Configuration Space—KT (Host)

## 52.5.1 Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 0h

**Default:** 19D58086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19d5<br>RO/V | **Device ID (DID):** This field identifies the particular device. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies the manufacturer of the device.<br>Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness.<br>The value of 0x8086 indicates Intel |

## 52.5.2 Status and Command (KT_HOST_STS_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 4h

**Default:** 00B00000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented. Hardwired to 0. |
| 30 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented. Hardwired to 0. |
| 29 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit must be set by a master device whenever its transaction (except for Special Cycle) is completed with Unsupported Request Completion Status (a.k.a. Master-Abort).<br>All master devices must implement this bit. |
| 28 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit must be set by a master device whenever its transaction is completed with Completer Abort Completion Status (a.k.a. Target-Abort).<br>All master devices must implement this bit. |
| 27 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit must be set by a target device whenever it completes a Posted or Non-Posted transaction with a Completer Abort (a.k.a. Target-Abort) error.<br>Devices that will never signal Completer Abort (a.k.a. Target-Abort) do not need to implement this bit. |
| 26:25 | 0x0<br>RO | **Devsel Timing (DEVT):** These bits encode the timing of DEVSEL#.<br>There are three allowable timings for assertion of DEVSEL# as described below:<br>• 00b: fast<br>• 01b: medium<br>• 10b: slow<br>• 11b: reserved<br>These bits are read-only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write.<br>Hardwired to 00b. |
| 24 | 0x0<br>RO | **Master Data Parity Error (MDPE):** Not implemented. Hardwired to 0. |
| 23 | 0x1<br>RO | **Fast Back to Back Capable (FBTBC):** This bit indicates whether or not the target is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.<br>This bit can be set to 1 if the device can accept these transactions and must be set to 0 otherwise.<br>Hardwired to 1. |
| 22 | 0x0<br>RO | **Reserved (RSVD3):** Reserved (RSVD) |
| 21 | 0x1<br>RO | **66 Mhz Capable (MCAP):** This bit indicates whether or not this device is capable of running at 66 MHz.<br>• A value of 0 indicates 33 MHz.<br>• A value of 1 indicates that the device is 66 MHz capable.<br>Hardwired to 1. |
| 20 | 0x1<br>RO | **Capabilities List (CAPL):** This optional read-only bit indicates whether or not this device implements the pointer for a New Capabilities linked list at offset 34h.<br>A value of zero indicates that no New Capabilities linked list is available.<br>A value of one indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 19 | 0x0<br>RO | **Interrupt Status (INTS):** This read-only bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted.<br>Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.<br>Read-only and hardwired to 0 for a device that does NOT support pin-based interrupt. |
| 18:16 | 0x0<br>RO | **Reserved (RSVD2):** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD1):** |
| 10 | 0x0<br>RW | **Interrupt Disable (INTD):** This bit disables the device/function from asserting INTx#.<br>A value of 0 enables the assertion of its INTx# signal.<br>A value of 1 disables the assertion of its INTx# signal.<br>This bit's state after RST# is 0. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBTBEN):** Not implemented.<br>Hardwired to 0. |
| 8 | 0x0<br>RO | **System Error Enable (SERREN):** Not implemented.<br>Hardwired to 0. |
| 7 | 0x0<br>RO | **Reserved (RSVD0):** |
| 6 | 0x0<br>RO | **Parity Error Response (PERRR):** Not implemented.<br>Hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** Not implemented.<br>Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWRIEN):** Not implemented.<br>Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycles (SPCYC):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the ability of a PCI device to issue Memory and I/O Read/Write Requests, and the ability of a PCI bridge to forward Memory and I/O Read/Write Requests in the Upstream direction.<br>When this bit is Set, the PCI device function is allowed to issue Memory or I/O Requests.<br>When this bit is Clear, the PCI device function is not allowed to issue any Memory or I/O Requests.<br>***Note:*** As MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well.<br>Requests other than Memory or I/O Requests are not controlled by this bit.<br>Default value of this bit is 0b.<br>This bit is hardwired to 0b if a PCI device function does not generate Memory or I/O Requests.<br>Default value of this bit is 0b. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls a device's response to Memory Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to Memory Space accesses. State after RST# is 0. |
| 0 | 0x0<br>RW | **IO Space Enable (IOSE):** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. |

## 52.5.3 Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 8h

**Default:** 07000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Identifies the Base Class Code of an external 16550-compatible serial controller device driver. |
| 23:16 | 0x0<br>RO | **Sub-Class Code (SCC):** Identifies the Sub-Class Code of an external 16550-compatible serial controller device driver. |
| 15:8 | 0x2<br>RO | **Programming Interface (PI):** Identifies the Programming Interface of an external 16550-compatible serial controller device driver. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** This register specifies a device specific revision identifier.<br>The value is chosen by the vendor.<br>Zero is an acceptable value.<br>This field should be viewed as a vendor defined extension to the Device ID. |

## 52.5.4 BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Built In Self Test (BIST):** Not implemented.<br>Hardwired to 0. |
| 23 | 0x1<br>RO/V | **Header Type 1 (HTYPE1):** This bit identifies whether or not the device contains multiple functions.<br>• If the bit is 0, then the device is single function.<br>• If the bit is 1, then the device has multiple functions. |
| 22:16 | 0x0<br>RO | **Header Type 0 (HTYPE0):** This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space).<br>• The encoding 00h specifies the non-bridge Configuration Space Header.<br>• The encoding 01h specifies the PCI-to-PCI bridge Configuration Space Header.<br>• The encoding 02h specifies the CardBus bridge Configuration Space Header.<br>• All other encodings are reserved.<br>Hardwired to 0 to identify the non-bridge Configuration Space Header. |
| 15:8 | 0x0<br>RO | **Latency Timer (LT):** Not implemented.<br>Hardwired to 0. |
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented.<br>Hardwired to 0. |

## 52.5.5 KT IO BAR (KT_HOST_IOBAR)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region |
| 2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.5.6 KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region |
| 11:4 | 0x0<br>RO | **Memory Size (MEMSIZE):** Hardwired to 0 to indicate 4KB of memory space |
| 3 | 0x0<br>RO | **Prefetchable (PREFETCH):** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.<br>Hardwired to 0 to indicate the device's memory space as non-prefetchable. |
| 2:1 | 0x0<br>RO | **Type (TYP):** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space. |
| 0 | 0x0<br>RO | **Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR. |

## 52.5.7 Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Cardbus CIS Pointer (CCP):** Not implemented. Hardwired to 0. |

## 52.5.8 Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch

These registers are used to uniquely identify the add-in card or subsystem where the PCI device resides. They provide a mechanism for add-in card vendors to distinguish their add-in cards from one another even though the add-in cards may have the same PCI controller on them (and, therefore, the same Vendor ID and Device ID).

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 2Ch

**Default:** 00008086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SID):** Value of this field is vendor-specific.<br>This is written by BIOS. No hardware action will be taken on this value.<br><br>Implementation Note: The Write-Once lock must be implemented per field. SID should have its own lock. |
| 15:0 | 0x8086<br>RW/O | **Subsystem Vendor ID (SVID):** This field identifies the vendor of the add-in card or subsystem.<br>This is written by BIOS. No hardware action will be taken on this value.<br><br>Implementation Note: The Write-Once lock must be implemented per field. SVID should have its own lock. |

## 52.5.9 Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Expansion ROM Base Address (XRBAR):** Not implemented. Hardwired to 0. |

## 52.5.10 Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 34h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** |
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list which is the MSI Capability. |

## 52.5.11 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Maximum Latency (MAXL):** Not implemented. Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Minimum Grant (MING):** Not implemented. Hardwired to 0. |
| 15:8 | 0x0<br>RO/V | **Interrupt Pin (INTP):** This register specifies which interrupt pin KT uses in PCI interrupt mode.<br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved. |
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information. POST software will write the routing information into this register as it initializes and configures the system.<br>The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to.<br>The device itself does not use this value, rather it is used by device drivers and operating systems.<br>Device drivers and operating systems can use this information to determine priority and vector information.<br>Values in this register are system architecture specific. |

## 52.5.12 MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 40h

**Default:** 00805005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD):** |
| 24 | 0x0<br>RO | **Per Vector Masking Capable (PVMC):** Not implemented.<br>Hardwired to 0 to indicate the function does NOT support MSI per-vector masking. |
| 23 | 0x1<br>RO | **64 bit address capable (XAC):** Hardwired to 1 to indicate the function is capable of sending a 64-bit message address. |
| 22:20 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW.<br>This field is RW for software compatibility, but only one interrupt vector is ever sent by the function. |
| 19:17 | 0x0<br>RO | **Multiple Message Capable (MMC):** Encoded number of interrupt vectors requested by a device.<br>Hardwired to 0 to indicate one requested interrupt vector. |
| 16 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI interrupt delivery is enabled whereas pin-based interrupt delivery SHALL be disabled.<br>If cleared, prior to returning the configuration write (that clears this field) completion, the function must send any pending MSI(s). |
| 15:8 | 0x50<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 50h to point to the PM Capability list |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Hardwired to 05h to indicate the linked list item as the MSI Capability registers |

## 52.5.13 MSI Message Address (KT_HOST_MSIMA)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** 32-bit DW-aligned MSI message address. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** |

## 52.5.14 MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Upper 32-bit of a 64-bit DW-aligned MSI message address. |

## 52.5.15 MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** |
| 15:0 | 0x0<br>RW | **Message Data (MD):** MSI Message Data |

## 52.5.16 Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 50h

**Default:** 00230001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#.<br>A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state.<br>• bit(27) X XXX1b - PME# can be asserted from D0<br>• bit(28) X XX1Xb - PME# can be asserted from D1<br>• bit(29) X X1XXb - PME# can be asserted from D2<br>• bit(30) X 1XXXb - PME# can be asserted from D3hot<br>• bit(31) 1 XXXXb - PME# can be asserted from D3cold |
| 26 | 0x0<br>RO | **D2 Support (D2S):** Hardwired to 0 to indicate that this device does not support D2 |
| 25 | 0x0<br>RO | **D1 Support (D1S):** Hardwired to 0 to indicate that this device does not support D1 |
| 24:22 | 0x0<br>RO | **Aux Current (AUXC):** Not implemented.<br>Hardwired to 0. |
| 21 | 0x1<br>RO | **Device Specific Initialization (DSI):** indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.<br>A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state.<br>Hardwired to 1 to indicate Device Specific Initialization is required. |
| 20 | 0x0<br>RO | **Reserved (RSVD):** |
| 19 | 0x0<br>RO | **PME Clock (PMECLK):** Not implemented.<br>Hardwired to 0. |
| 18:16 | 0x3<br>RO | **Version (VER):** Hardwired to value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec |
| 15:8 | 0x0<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 0 to indicate no more linked list item. |
| 7:0 | 0x1<br>RO | **Capability ID (CID):** Hardwired to 01h to indicate the linked list item as the PCI Power Management registers |

## 52.5.17 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + 54h

**Default:** 00000008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Data:** Not implemented.<br>Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Control/Status Register Bridge Support Extensions (CSRBSE):** Not implemented.<br>Hardwired to 0. |
| 15 | 0x0<br>RO | **PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit.<br>• Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled).<br>• Writing a 0 has no effect.<br><br>If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.<br><br>Not implemented. Hardwired to 0. |
| 14:13 | 0x0<br>RO | **Data Scale (DS):** Not implemented.<br>Hardwired to 0. |
| 12:9 | 0x0<br>RO | **Data Select (DSEL):** Not implemented.<br>Hardwired to 0. |
| 8 | 0x0<br>RO | **PME Enable (PMEEN):** A 1 enables the function to assert PME. When 0, PME assertion is disabled.<br>This bit defaults to 0 if the function does not support PME generation from D3cold.<br>If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.<br><br>Not implemented. Hardwired to 0. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br>Configuration Context is preserved.<br>Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. |
| 2 | 0x0<br>RO | **Reserved (RSVD0):** |
| 1:0 | 0x0<br>RW | **Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state.<br>The definition of the field values is given below:<br>• 00b - D0<br>• 01b - D1<br>• 10b - D2<br>• 11b - D3hot<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. |

## 52.5.18 Process Stepping Information (KT_HOST_PSI)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:3] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Reserved. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping (MSID):** Reserved. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** Reserved. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Reserved. |

## 52.6 Registers in Configuration Space—IDE (Host)

**AR Author and Tech Comm:** HIDE this sub-chapter if the device has no such registers.

### 52.6.1 Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 0h

**Default:** 19EA8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19ea<br>RO/V | **Device ID (DID):** This field identifies the particular device. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies the manufacturer of the device.<br>Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness.<br>The value of 0x8086 indicates Intel |

## 52.6.2 Status and Command (IDE_HOST_STS_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 4h

**Default:** 00B00000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented.<br>Hardwired to 0. |
| 30 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented.<br>Hardwired to 0. |
| 29 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit must be set by a master device whenever its transaction (except for Special Cycle) is completed with Unsupported Request Completion Status (a.k.a. Master-Abort).<br>All master devices must implement this bit. |
| 28 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit must be set by a master device whenever its transaction is completed with Completer Abort Completion Status (a.k.a. Target-Abort).<br>All master devices must implement this bit. |
| 27 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit must be set by a target device whenever it completes a Posted or Non-Posted transaction with a Completer Abort (a.k.a. Target-Abort) error.<br>Devices that will never signal Completer Abort (a.k.a. Target-Abort) do not need to implement this bit. |
| 26:25 | 0x0<br>RO | **Devsel Timing (DEVT):** These bits encode the timing of DEVSEL#.<br>There are three allowable timings for assertion of DEVSEL# as described below:<br>• 00b: fast;<br>• 01b: medium;<br>• 10b: slow;<br>• 11b: reserved.<br>These bits are read-only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write.<br><br>Hardwired to 00b. |
| 24 | 0x0<br>RO | **Master Data Parity Error (MDPE):** Not implemented.<br>Hardwired to 0. |
| 23 | 0x1<br>RO | **Fast Back to Back Capable (FBTBC):** This bit indicates whether or not the target is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.<br>This bit can be set to 1 if the device can accept these transactions and must be set to 0 otherwise.<br>Hardwired to 1. |
| 22 | 0x0<br>RO | **Reserved (RSVD3):** Reserved (RSVD) |
| 21 | 0x1<br>RO | **66 Mhz Capable (MCAP):** This bit indicates whether or not this device is capable of running at 66 MHz.<br>• A value of 0 indicates 33 MHz.<br>• A value of 1 indicates that the device is 66 MHz capable.<br><br>Hardwired to 1. |
| 20 | 0x1<br>RO | **Capabilities List (CAPL):** This optional read-only bit indicates whether or not this device implements the pointer for a New Capabilities linked list at offset 34h.<br>A value of zero indicates that no New Capabilities linked list is available.<br>A value of one indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 19 | 0x0<br>RO | **Interrupt Status (INTS):** This read-only bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted.<br>Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.<br>Read-only and hardwired to 0 for a device that does NOT support pin-based interrupt. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 18:16 | 0x0<br>RO | **Reserved (RSVD2):** |
| 15:11 | 0x0<br>RO | **Reserved (RSVD1):** |
| 10 | 0x0<br>RW | **Interrupt Disable (INTD):** This bit disables the device/function from asserting INTx#.<br>• A value of 0 enables the assertion of its INTx# signal.<br>• A value of 1 disables the assertion of its INTx# signal.<br>This bit's state after RST# is 0. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBTBEN):** Not implemented.<br>Hardwired to 0. |
| 8 | 0x0<br>RO | **System Error Enable (SERREN):** Not implemented.<br>Hardwired to 0. |
| 7 | 0x0<br>RO | **Reserved (RSVD0):** |
| 6 | 0x0<br>RO | **Parity Error Response (PERRR):** Not implemented.<br>Hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** Not implemented.<br>Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWRIEN):** Not implemented.<br>Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycles (SPCYC):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the ability of a PCI device to issue Memory and I/O Read/Write Requests, and the ability of a PCI bridge to forward Memory and I/O Read/Write Requests in the Upstream direction.<br>When this bit is Set, the PCI device function is allowed to issue Memory or I/O Requests.<br>When this bit is Clear, the PCI device function is not allowed to issue any Memory or I/O Requests.<br>***Note:*** As MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well.<br>Requests other than Memory or I/O Requests are not controlled by this bit.<br>Default value of this bit is 0b.<br>This bit is hardwired to 0b if a PCI device function does not generate Memory or I/O Requests.<br>Default value of this bit is 0b. |
| 1 | 0x0<br>RO | **Memory Space Enable (MSE):** Controls a device's response to Memory Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to Memory Space accesses. State after RST# is 0.<br>Read-only and hardwired to 0 because IDE does NOT support Memory Space accesses. |
| 0 | 0x0<br>RW | **IO Space Enable (IOSE):** Controls a device's response to I/O Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to I/O Space accesses.<br>State after RST# is 0. |

## 52.6.3 Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 8h

**Default:** 01018500h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x1<br>RO | **Base Class Code (BCC):** Identifies the Base Class Code of an external IDE controller device driver. |
| 23:16 | 0x1<br>RO | **Sub-Class Code (SCC):** Identifies the Sub-Class Code of an external IDE controller device driver. |
| 15:8 | 0x85<br>RO | **Programming Interface (PI):** Identifies the Programming Interface of an IDE controller device driver. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** This register specifies a device specific revision identifier.<br>The value is chosen by the vendor.<br>Zero is an acceptable value.<br>This field should be viewed as a vendor defined extension to the Device ID. |

## 52.6.4 BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Built In Self Test (BIST):** Not implemented. Hardwired to 0. |
| 23 | 0x1<br>RO/V | **Header Type 1 (HTYPE1):** This bit identifies whether or not the device contains multiple functions.<br>• If the bit is 0, then the device is single function.<br>• If the bit is 1, then the device has multiple functions. |
| 22:16 | 0x0<br>RO | **Header Type 0 (HTYPE0):** This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space).<br>• The encoding 00h specifies the non-bridge Configuration Space Header.<br>• The encoding 01h specifies the PCI-to-PCI bridge Configuration Space Header.<br>• The encoding 02h specifies the CardBus bridge Configuration Space Header.<br>• All other encodings are reserved.<br><br>Hardwired to 0 to identify the non-bridge Configuration Space Header. |
| 15:8 | 0x0<br>RO | **Latency Timer (LT):** Not implemented. Hardwired to 0. |
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented. Hardwired to 0. |

## 52.6.5 IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region |
| 2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.6.6 IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 14h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.6.7 IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 18h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region |
| 2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.6.8 IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 1Ch

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.6.9 IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 20h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 3:2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 16B of IO space. |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 52.6.10 Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Cardbus CIS Pointer (CCP):** Not implemented. Hardwired to 0. |

## 52.6.11 Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch

These registers are used to uniquely identify the add-in card or subsystem where the PCI device resides. They provide a mechanism for add-in card vendors to distinguish their add-in cards from one another even though the add-in cards may have the same PCI controller on them (and, therefore, the same Vendor ID and Device ID).

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 2Ch

**Default:** 00008086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SID):** Value of this field is vendor-specific.<br>This is written by BIOS. No hardware action will be taken on this value.<br><br>Implementation Note: The Write-Once lock must be implemented per field. SID should have its own lock. |
| 15:0 | 0x8086<br>RW/O | **Subsystem Vendor ID (SVID):** This field identifies the vendor of the add-in card or subsystem.<br>This is written by BIOS. No hardware action will be taken on this value.<br><br>Implementation Note: The Write-Once lock must be implemented per field. SVID should have its own lock. |

## 52.6.12 Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Expansion ROM Base Address (XRBAR):** Not implemented.<br>Hardwired to 0. |

## 52.6.13 Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 34h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** |
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list which is the MSI Capability. |

## 52.6.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Maximum Latency (MAXL):** Not implemented.<br>Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Minimum Grant (MING):** Not implemented.<br>Hardwired to 0. |
| 15:8 | 0x0<br>RO/V | **Interrupt Pin (INTP):** This register specifies which interrupt pin IDE uses in PCI interrupt mode.<br>Value Decoding<br>00h The function does NOT use an interrupt pin.<br>01h INTA<br>02h INTB<br>03h INTC<br>04h INTD<br>05h - FFh Reserved.<br><br>***Note:*** This field shadows the IDEHIPINR.IPIN field in the PTIO Host private CR space which is configured by BIOS over IOSF SB. |
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information.<br>POST software will write the routing information into this register as it initializes and configures the system.<br>The value in this register tells which input of the system interrupt controller(s) the device's interrupt pin is connected to.<br>The device itself does not use this value, rather it is used by device drivers and operating systems.<br>Device drivers and operating systems can use this information to determine priority and vector information.<br>Values in this register are system architecture specific. |

## 52.6.15 MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 40h

**Default:** 00805005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD):** |
| 24 | 0x0<br>RO | **Per Vector Masking Capable (PVMC):** Not implemented.<br>Hardwired to 0 to indicate the function does NOT support MSI per-vector masking. |
| 23 | 0x1<br>RO | **64 bit address capable (XAC):** Hardwired to 1 to indicate the function is capable of sending a 64-bit message address. |
| 22:20 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW.<br>This field is RW for software compatibility, but only one interrupt vector is ever sent by the function. |
| 19:17 | 0x0<br>RO | **Multiple Message Capable (MMC):** Encoded number of interrupt vectors requested by a device.<br>Hardwired to 0 to indicate one requested interrupt vector. |
| 16 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI interrupt delivery is enabled whereas pin-based interrupt delivery SHALL be disabled.<br>If cleared, prior to returning the configuration write (that clears this field) completion, the function must send any pending MSI(s). |
| 15:8 | 0x50<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 50h to point to the PM Capability list |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Hardwired to 05h to indicate the linked list item as the MSI Capability registers |

## 52.6.16 MSI Message Address (IDE_HOST_MSIMA)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** 32-bit DW-aligned MSI message address. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** |

## 52.6.17 MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Upper 32-bit of a 64-bit DW-aligned MSI message address. |

## 52.6.18 MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** |
| 15:0 | 0x0<br>RW | **Message Data (MD):** MSI Message Data |

## 52.6.19 Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 50h

**Default:** 00230001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state.<br>• bit(27) X XXX1b - PME# can be asserted from D0<br>• bit(28) X XX1Xb - PME# can be asserted from D1<br>• bit(29) X X1XXb - PME# can be asserted from D2<br>• bit(30) X 1XXXb - PME# can be asserted from D3hot<br>• bit(31) 1 XXXXb - PME# can be asserted from D3cold |
| 26 | 0x0<br>RO | **D2 Support (D2S):** Hardwired to 0 to indicate that this device does not support D2 |
| 25 | 0x0<br>RO | **D1 Support (D1S):** Hardwired to 0 to indicate that this device does not support D1 |
| 24:22 | 0x0<br>RO | **Aux Current (AUXC):** Not implemented.<br>Hardwired to 0. |
| 21 | 0x1<br>RO | **Device Specific Initialization (DSI):** indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.<br>A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state.<br>Hardwired to 1 to indicate Device Specific Initialization is required. |
| 20 | 0x0<br>RO | **Reserved (RSVD):** |
| 19 | 0x0<br>RO | **PME Clock (PMECLK):** Not implemented.<br>Hardwired to 0. |
| 18:16 | 0x3<br>RO | **Version (VER):** Hardwired to value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec |
| 15:8 | 0x0<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 0 to indicate no more linked list item. |
| 7:0 | 0x1<br>RO | **Capability ID (CID):** Hardwired to 01h to indicate the linked list item as the PCI Power Management registers |

## 52.6.20 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + 54h

**Default:** 00000008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Data:** Not implemented.<br>Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Control/Status Register Bridge Support Extensions (CSRBSE):** Not implemented.<br>Hardwired to 0. |
| 15 | 0x0<br>RO | **PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit.<br>• Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled).<br>• Writing a 0 has no effect.<br><br>If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.<br><br>Not implemented. Hardwired to 0. |
| 14:13 | 0x0<br>RO | **Data Scale (DS):** Not implemented.<br>Hardwired to 0. |
| 12:9 | 0x0<br>RO | **Data Select (DSEL):** Not implemented.<br>Hardwired to 0. |
| 8 | 0x0<br>RO | **PME Enable (PMEEN):** A 1 enables the function to assert PME.<br>When 0, PME assertion is disabled.<br>This bit defaults to 0 if the function does not support PME generation from D3cold.<br>If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.<br><br>Not implemented. Hardwired to 0. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br>Configuration Context is preserved.<br>Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. |
| 2 | 0x0<br>RO | **Reserved (RSVD0):** |
| 1:0 | 0x0<br>RW | **Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state.<br>The definition of the field values is given below:<br>00b - D0<br>01b - D1<br>10b - D2<br>11b - D3hot<br><br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. |

## 52.6.21 Process Stepping Information (IDE_HOST_PSI)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:24, F:2] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Reserved. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping (MSID):** Reserved. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** Reserved. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Reserved. |

## 52.7 Registers in Memory Space—HECI1_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI1_MMIO_MBAR_LO described in Section 52.2.9, "HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h" on page 2243 and HECI1_MMIO_MBAR_HI described in Section 52.2.10, "HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h" on page 2244.

### 52.7.1 Host CB Write Window (HECI1_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI1_MMIO_MBAR] + 0h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:24, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data.<br>Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 52.7.2 Host Control and Status Register (HECI1_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:24, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br>Programmers note: This field is implemented with a '1-hot' scheme. Only one bit will be set to a '1' at a time. Each bit position represents the value n of a buffer depth of (2^n). For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA. HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 52.7.3 CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:24, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 52.7.4 CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:24, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD. **This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 52.7.5 D0i3 Control (HECI1_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 800h

**Default:** 00000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:24, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 52.8 Registers in Memory Space—HECI2_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI2_MMIO_MBAR_LO described in Section 52.3.9, "HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h" on page 2263 and HECI2_MMIO_MBAR_HI described in Section 52.3.10, "HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h" on page 2264.

### 52.8.1 Host CB Write Window (HECI2_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI2_MMIO_MBAR] + 0h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:24, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data. Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 52.8.2 Host Control and Status Register (HECI2_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:24, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br><br>Programmers note: This field is implemented with a '1-hot' scheme. Only one bit will be set to a '1' at a time. Each bit position represents the value n of a buffer depth of (2^n). For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA.<br>HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 52.8.3 CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:24, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 52.8.4 CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:24, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD. **This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 52.8.5 D0i3 Control (HECI2_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI2_MMIO_MBAR] + 800h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:24, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 52.9 Registers in Memory Space—HECI3_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI3_MMIO_MBAR_LO described in Section 52.4.9, "HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h" on page 2280 and HECI3_MMIO_MBAR_HI described in Section 52.4.10, "HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h" on page 2281.

### 52.9.1 Host CB Write Window (HECI3_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI3_MMIO_MBAR] + 0h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:24, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data.<br>Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 52.9.2 Host Control and Status Register (HECI3_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:24, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br>Programmers note: This field is implemented with a '1-hot' scheme. Only one bit will be set to a '1' at a time. Each bit position represents the value n of a buffer depth of (2^n). For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position. H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA. HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position. H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 52.9.3 CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:24, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 52.9.4 CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:24, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD. **This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 52.9.5 D0i3 Control (HECI3_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI3_MMIO_MBAR] + 800h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:24, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 52.10 Registers in Memory Space—KT_HOST_MEMBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) KT_HOST_MEMBAR described in Section 52.5.6, "KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h" on page 2296.

### 52.10.1 KT Receive Buffer Register (MEM_KTRBR)—Offset 0h

This register implements the standard Receiver Buffer register in the KT Block. This is read-only. DLAB (i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be read when the receiver data is ready. In FIFO mode, a read from this register will be translated into a read from the RBR FIFO.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Receiver Buffer Register (RBR):** Implements the Receiver Buffer register of the Serial Interface. |

### 52.10.2 KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h

This register implements the standard Transmitter Holding register in the KT Block. This is write-only. DLAB (i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be written when the Host needs to transmit data. In FIFO mode, a write to this register will be translated into a write to the THR FIFO.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **Transmitter Holding Register (THR):** Implements the Transmit Data register of the Serial Interface. |

## 52.10.3 KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h

This register implements the standard DLL register in the KT Block. DLAB (i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. this register is supported only for SW compatibility and has no impact on HW performance.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch LSB (DLL):** Implements the DLL register of the Serial Interface. |

## 52.10.4 KT Interrupt Enable register (MEM_KTIER)—Offset 1h

This register implements the standard Interrupt Enable register in the KT Block. The bits enable specific events to interrupt the Host. See bit specific definition below. DLAB (i.e. KTLCR[7]) shall is 0 in order for this register to be accessed.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 1h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RW | **MSR (IER3):** When set, enables bits in Modem Status register to cause an interrupt to the Host |
| 2 | 0x0<br>RW | **LSR (IER2):** When set, enables bits in Receiver Line Status Register to cause an interrupt to the Host |
| 1 | 0x0<br>RW | **THR (IER1):** When set, enables an interrupt to the Host when the Transmitter Holding register is empty |
| 0 | 0x0<br>RW | **DR (IER0):** When set, enables an interrupt to the Host when the Received Data is available OR Receiver FIFO timeout. |

## 52.10.5 KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h

This register implements the standard DLM register in the KT Block. DLAB (i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. This register is supported only for SW compatibility and has no impact on HW performance.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 1h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch MSB (DLM):** Implements the DLM register of the Serial Interface. |

## 52.10.6 KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h

This register implements the standard Interrupt Identification register in the KT Block. This is read-only. This register is created to provide minimum software overhead during data character transfers. The register prioritizes KT interrupts into 4 levels and records them in the IIRSTS field of this register. The four levels of interrupt conditions in order of priority are: Receiver Line Status; Received Data Ready/Character Timeout Indication; Transmitter Holding Register Empty; MODEM Status. When the Host accesses the IIR, the HW freezes all interrupts and indicates the highest priority pending interrupt to the Host. While this Host access is occurring, the HW records new interrupts, but does not change its current indication until the access is complete. Table in the Host Interrupt Generation section shows the contents.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 2h

**Default:** 01h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **FIFO Enable (FIEN1):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 6 | 0x0<br>RO/V | **FIFO Enable (FIEN0):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0. |
| 3:1 | 0x0<br>RO/V | **IIR Status (IIRSTS):** These bits are asserted by the HW according to the source of the interrupt and the priority level.<br>Refer to the section on Host Interrupt Generation for a table of values. |
| 0 | 0x1<br>RO/V | **Interrupt Status (INTSTS):** 0: indicates a pending Host interrupt and IIRSTS may be used as a pointer to the appropriate interrupt service routine. It reflects the logical-NOR of all UART interrupts. An individual UART interrupt will be identified as pending when the individual event status is asserted AND its corresponding event interrupt enable is asserted as well.<br>For example, the THR Empty interrupt will be identified as pending when both LSR[THRE] bit is 1 AND IER[IER1] is 1.<br>1: indicates no pending Host interrupt. |

## 52.10.7 KT FIFO Control register (MEM_KTFCR)—Offset 2h

This register implements the standard FIFO Control register in the KT Block. It is used to configure features pertinent to the FIFO mode of operation. This is write-only.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 2h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Receiver Trigger Level (RTL):** Trigger level in bytes for the Receiver FIFO.<br>Once the trigger level number of bytes is reached, an interrupt will be sent to the Host [7:6] RCVR FIFO Trigger Level (bytes)<br>• 00 -- 01<br>• 01 -- 04<br>• 10 -- 08<br>• 11 -- 14 |
| 5:4 | 0x0<br>WO | **Reserved (RSVD):** Writes to this field shall be dropped. |
| 3 | 0x0<br>WO | **RDY Mode (RDYM):** This bit has no effect on HW performance |
| 2 | 0x0<br>WO | **XMT FIFO Clear (XFIC):** When the Host writes one to this bit the HW will clear the XMT FIFO.<br>This bit is self-cleared by HW. |
| 1 | 0x0<br>WO | **RCV FIFO Clear (RFIC):** When the Host writes one to this bit the HW will clear the RCV FIFO.<br>This bit is self-cleared by HW. |
| 0 | 0x0<br>WO | **FIFO Enable (FIE):** When set indicates that the KT interface is working in FIFO node.<br>When this bit value is changed the RCV and XMT FIFO are cleared by HW.<br>This bit must be a 1 when other FCR bits are written to or they will NOT be programmed. |

## 52.10.8 KT Line Control register (MEM_KTLCR)—Offset 3h

This register implements the standard Line Control register in the KT Block. It specifies the format of the asynchronous data communications exchange and sets the DLAB bit. Most bits in this register have no effect on HW and are only used by the FW.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 3h

**Default:** 03h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Divisor Latch Access Bit (DLAB):**<br>• 1: Allows access to the Divisor Latch Registers and reading of the FIFO Control Register.<br>• 0: Allows access to RBR, THR, IER and IIR registers. |
| 6 | 0x0<br>RW | **Break Control (BC):** This bit has no effect on HW |
| 5:4 | 0x0<br>RW | **Parity Bit Mode (PBM):** This bit has no effect on HW |
| 3 | 0x0<br>RW | **Parity Enable (PE):** This bit has no effect on HW |
| 2 | 0x0<br>RW | **Stop Bit Select (SBS):** This bit has no effect on HW |
| 1:0 | 0x3<br>RW | **Word Select Byte (WSB):** This bit has no effect on HW. |

## 52.10.9 KT Modem Control register (MEM_KTMCR)—Offset 4h

This register implements the standard Modem Control register in the KT Block. It controls the interface with the modem. Because the FW emulates the modem, the Host communicates with the FW via this register. This register has impact on the HW in the loopback mode.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 4h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4 | 0x0<br>RW | **Loop Back Mode (LBM):** When set by Host indicates that the serial port is in loop Back mode. This means that the data that is transmitted by the host should be received. Helps in debug of the interface |
| 3 | 0x0<br>RW | **Output 2 (OUT2):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit (7) |
| 2 | 0x0<br>RW | **Output 1 (OUT1):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit(6) |
| 1 | 0x0<br>RW | **Request to Send Out (RTSO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value of this bit is written by HW to Modem Status Register bit(4) |
| 0 | 0x0<br>RW | **Data Terminal Ready Out (DRTO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value in this bit is written by HW to Modem Status Register Bit(5) |

## 52.10.10 KT Line Status register (MEM_KTLSR)—Offset 5h

This register implements the standard Line Status register in the KT Block. It provides status information of the data transfer to the Host. Error indication etc are provided by the HW/FW to the host via this register.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 5h

**Default:** 60h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/C/V | **RCVR FIFO Error (RXFER):** This bit is cleared in non FIFO mode.<br>This bit is connected to UART LSR[BI] bit in FIFO mode. |
| 6 | 0x1<br>RO/V | **Transmitter Empty (TEMT):** In non-FIFO mode:<br>this bit will be set to 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, OR cleared to 0 whenever either THR or TSR contains a data byte.<br><br>In FIFO mode:<br>this bit will be set to 1 whenever XMIT FIFO and TSR are both empty, OR cleared to 0 whenever either XMIT FIFO or TSR contains a data byte.<br><br>Implementation note: This bit is connected by HW to THRE bit |
| 5 | 0x1<br>RO/V | **Transmitter Holding Register Empty (THRE):** The bit will always be set whenever Host changes UART FCR[FIE] bit.<br>This bit is controlled differently in FIFO vs non-FIFO modes:<br>Non-FIFO mode: This bit will be cleared by HW when Host writes to THR, and set by HW when FW reads from THR.<br>FIFO mode: This bit will be set by HW when XMIT FIFO is empty , and cleared by HW when XMIT FIFO is NOT empty. |
| 4 | 0x0<br>RO/C/V | **Break Interrupt (BI):** This bit will be cleared by HW when the UART LSR register is read by Host.<br>This bit will be set by HW in two cases:<br>In FIFO mode, when FW sets the KTRIVR[SBI] bit.<br>In non-FIFO mode, when FW sets the KTRBR[BIA] bit. |
| 3 | 0x0<br>RO | **Framing Error (FE):** This bit is NOT implemented and hardwired to 0. |
| 2 | 0x0<br>RO | **Parity Error (PE):** This bit is NOT implemented and hardwired to 0. |
| 1 | 0x0<br>RO/C/V | **Overrun Error (OE):** HW will clear this bit when Host reads from KTLSR register and this is source of interrupt.<br>HW will set this bit when overrun error is detected.<br>In non-FIFO mode, the character that triggers the overrun error shall overwrite the character in the RBR register.<br>In FIFO mode, the character that triggers the overrun error shall be dropped without corrupting the RCVR FIFO. |
| 0 | 0x0<br>RO/V | **Data Ready (DR):** Non-FIFO Mode: this bit will be set by HW when FW writes to RBR and cleared by HW when Host reads from RBR.<br>FIFO Mode: the bit will be set by HW when RCVR FIFO is not empty and cleared by HW when RCVR FIFO is empty. |

## 52.10.11 KT Modem Status register (MEM_KTMSR)—Offset 6h

This register implements the standard Modem Status register in the KT Block. It provides the current state of the control lines from the modem. The modem functionality is emulated by the FW.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 6h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Data Carrier Detect (DCD):** In Loop Back mode this bit is connected by HW to the value of MCR bit 3 |
| 6 | 0x0<br>RO | **Ring Indicator (RI):** In Loop Back mode this bit is connected by HW to the value of MCR bit 2. |
| 5 | 0x0<br>RO | **Data Set Ready (DSR):** In Loop Back mode this bit is connected by HW to the value of MCR bit 0. |
| 4 | 0x0<br>RO | **Clear To Send (CTS):** In Loop Back mode this bit is connected by HW to the value of MCR bit 1. |
| 3 | 0x0<br>RO/C/V | **Delta Data Carrier Detect (DDCD):** This bit is set when bit 7 is changed.<br>This bit is cleared by HW when the MSR register is being read by the HOST driver. |
| 2 | 0x0<br>RO/C/V | **Trailing Edge of Read Detector (TERI):** This bit is set when bit 6 is changed from 1 to 0.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 1 | 0x0<br>RO/C/V | **Delta Data Set Ready (DDSR):** This bit is set when bit 5 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver |
| 0 | 0x0<br>RO/C/V | **Delta Clear To Send (DCTS):** This bit is set when bit 4 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver |

## 52.10.12 KT Scratch register (MEM_KTSCR)—Offset 7h

This register implements the standard Scratch register in the KT Block. It allows Host SW programmers to hold temporary data. It has NO effect on HW.

Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 7h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:24, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Scratch Register Data (SCRD):** Reserved. |

## 52.11 Registers in Memory Space—Fixed Address

These registers are accessed in Host Memory Space through fixed memory addresses.

### 52.11.1 FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h

Reads by host SW from this register will reflect the value of FTPM_LOC_STATE register in mIA space. This register is aliased to the following addresses in the host address space: FED4_x000, x=0,1,2,3,4. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40000h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO/V | **Reserved (RSVD_31_8):** Opaque to HW. Defined by FW. |
| 7 | 0x0<br>RO/V | **Register Valid Status (FTPM_REG_VALID_STS):** Opaque to HW.<br>This bit is used by FW to signal to host SW when this register is valid. |
| 6:5 | 0x0<br>RO/V | **Reserved (RSVD_6_5):** Opaque to HW. Defined by FW. |
| 4:2 | 0x0<br>RO/V | **Active Locality (FTPM_ACTV_LOC):** The encodings are:<br>• 000 - Locality 0,<br>• 001 - Locality 1,<br>• 010 - Locality 2,<br>• 011 - Locality 3,<br>• 100 - Locality 4,<br>• 101 and 111 - reserved.<br>FW sets these bits to mux in the active locality.<br>Once set, the FTPM HCI will pass cycles from the active host locality to the FTPM shared SRAM.<br>Writes from in-active localities will be dropped and reads will return 0's. |
| 1 | 0x0<br>RO/V | **Locality Assigned (FTPM_LOC_ASSIGNED):** A value of 0 indicates to host SW that no locality is assigned, a value of 1 indicates a locality has been assigned. |
| 0 | 0x0<br>RO/V | **Establishment (FTPM_ESTABLISHMENT):** Opaque to HW.<br>This bit is set to 0 by FW when HASH_START occurs and set to 1 when the RESET_ESTABLISHMENT bit is set. |

### 52.11.2 FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h

This register was added in order to support QW reads from Locality State register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40004h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Reserved (FTPM_LOC_RSVD_BITS):** Reserved. |

### 52.11.3 FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h

Writes by host SW to this register will update the value of the corresponding LOC_n_CTRL_CSEROA register in mIA space. Writes by host SW to this register may be configured by CSE FW to trigger an MSI/PME to mIA. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40008h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **FTPM Locality n Control Bits (FTPM_LOC_CTRL_BITS):** Reserved. |

### 52.11.4 FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch

Reads by host SW from this register will reflect the value of the corresponding LOC_n_STS register in mIA space. This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4000Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Locality n Status Bits (FTPM_LOC_STS_BITS):** Reserved. |

### 52.11.5 FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. This DW is aliased to the same offset for each locality, and even inactive localities can access it.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-7]:** [0x0] + FED40010h + [0-7]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Locality Config Area Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

### 52.11.6 FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h

This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40030h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Interface ID Bits 0 (FTPM_INTF_ID_BITS_0):** These are the 32 LSbits used by FW to inform host SW as to the interface type of the TPM.<br>The exact usage is defined by TCG. |

### 52.11.7 FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h

This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40034h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Interface ID Bits 1 (FTPM_INTF_ID_BITS_1):** These are the 32 MSbits used by FW to inform host SW as to the interface type of the TPM.<br>The exact usage is defined by TCG. |

## 52.11.8 FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. This DW is aliased to the same offset for each locality, and even inactive localities can access it.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-1]:** [0x0] + FED40038h + [0-1]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Locality Config Area Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

## 52.11.9 Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40040h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_REQUEST:** Host software sets this field to a CSxE firmware-defined value to indicate a state change request to the fTPM.<br>Any host write to this register shall result in an interrupt to the CSxE firmware, or wakeup the CSxE subsystem when it is in power-gated state. |

## 52.11.10 Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40044h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_STATUS:** Reserved. |

## 52.11.11 Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40048h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CANCEL:** Host software sets this field to a CSxE firmware-defined value to indicate that a command should be canceled.<br>Any host write to this register shall result in an interrupt to the CSxE firmware, or wakeup the CSxE subsystem when it is in power-gated state. |

## 52.11.12 Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4004Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Control Area Start (HOST_CA_START):** Host software sets this field to a CSxE firmware defined value to indicate that a command is available for processing.<br>The reset value indicates that a command is NOT available for processing.<br>Any host write to this register shall result in an interrupt to the CSE firmware or wakeup the CSE subsystem when it is in power-gated state. |

## 52.11.13 Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40050h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RSVD_31_0:** Reserved |

## 52.11.14 Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40054h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RSVD_31_0:** Reserved |

## 52.11.15 Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40058h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD_SZ:** Reserved. |

## 52.11.16 Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4005Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD1:** Reserved. |

## 52.11.17 Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40060h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD2:** Reserved. |

## 52.11.18 Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40064h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP_SZ:** Reserved. |

## 52.11.19 Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40068h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP1:** Reserved. |

## 52.11.20 Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4006Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP2:** Reserved. |

## 52.11.21 FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. The access to this DW is allowed only to an active locality.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-3]:** [0x0] + FED40070h + [0-3]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

## 52.11.22 FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h

This register represents a DW of RW memory in FTPM CRB (EBB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-991]:** [0x0] + FED40080h + [0-991]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Command and Response Buffer DW (HOST_FTPM_CRB_DW):** Reserved. |

## 52.12 Registers in I/O Space—KT_HOST_IOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) KT_HOST_IOBAR described in Section 52.5.5, "KT IO BAR (KT_HOST_IOBAR)—Offset 10h" on page 2296.

### 52.12.1 KT Receive Buffer Register (IO_KTRBR)—Offset 0h

This register implements the standard Receiver Buffer register in the KT Block. This is read-only. DLAB (i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be read when the receiver data is ready. In FIFO mode, a read from this register will be translated into a read from the RBR FIFO.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Receiver Buffer Register (RBR):** Implements the Receiver Buffer register of the Serial Interface. |

### 52.12.2 KT Transmitter Holding Register (IO_KTTHR)—Offset 0h

This register implements the standard Transmitter Holding register in the KT Block. This is write-only. DLAB (i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be written when the Host needs to transmit data. In FIFO mode, a write to this register will be translated into a write to the THR FIFO.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **Transmitter Holding Register (THR):** Implements the Transmit Data register of the Serial Interface. |

## 52.12.3 KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h

This register implements the standard DLL register in the KT Block. DLAB (i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. this register is supported only for SW compatibility and has no impact on HW performance.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch LSB (DLL):** Implements the DLL register of the Serial Interface. |

## 52.12.4 KT Interrupt Enable register (IO_KTIER)—Offset 1h

This register implements the standard Interrupt Enable register in the KT Block. The bits enable specific events to interrupt the Host. See bit specific definition below. DLAB (i.e. KTLCR[7]) shall is 0 in order for this register to be accessed.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 1h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RW | **MSR (IER3):** When set, enables bits in Modem Status register to cause an interrupt to the Host |
| 2 | 0x0<br>RW | **LSR (IER2):** When set, enables bits in Receiver Line Status Register to cause an interrupt to the Host |
| 1 | 0x0<br>RW | **THR (IER1):** When set, enables an interrupt to the Host when the Transmitter Holding register is empty |
| 0 | 0x0<br>RW | **DR (IER0):** When set, enables an interrupt to the Host when the Received Data is available OR Receiver FIFO timeout. |

## 52.12.5 KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h

This register implements the standard DLM register in the KT Block. DLAB (i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. This register is supported only for SW compatibility and has no impact on HW performance.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 1h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch MSB (DLM):** Implements the DLM register of the Serial Interface. |

## 52.12.6 KT Interrupt Identification Register (IO_KTIIR)—Offset 2h

This register implements the standard Interrupt Identification register in the KT Block. This is read-only. This register is created to provide minimum software overhead during data character transfers. The register prioritizes KT interrupts into 4 levels and records them in the IIRSTS field of this register. The four levels of interrupt conditions in order of priority are: Receiver Line Status; Received Data Ready/Character Timeout Indication; Transmitter Holding Register Empty; MODEM Status. When the Host accesses the IIR, the HW freezes all interrupts and indicates the highest priority pending interrupt to the Host. While this Host access is occurring, the HW records new interrupts, but does not change its current indication until the access is complete. Table in the Host Interrupt Generation section shows the contents.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 01h

**BAR and Offset:** [KT_HOST_IOBAR] + 2h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **FIFO Enable (FIEN1):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 6 | 0x0<br>RO/V | **FIFO Enable (FIEN0):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0. |
| 3:1 | 0x0<br>RO/V | **IIR Status (IIRSTS):** These bits are asserted by the HW according to the source of the interrupt and the priority level.<br>Refer to the section on Host Interrupt Generation for a table of values. |
| 0 | 0x1<br>RO/V | **Interrupt Status (INTSTS):** 0: indicates a pending Host interrupt and IIRSTS may be used as a pointer to the appropriate interrupt service routine. It reflects the logical-NOR of all UART interrupts. An individual UART interrupt will be identified as pending when the individual event status is asserted AND its corresponding event interrupt enable is asserted as well.<br>For example, the THR Empty interrupt will be identified as pending when both LSR[THRE] bit is 1 AND IER[IER1] is 1.<br>1: indicates no pending Host interrupt. |

## 52.12.7 KT FIFO Control register (IO_KTFCR)—Offset 2h

This register implements the standard FIFO Control register in the KT Block. It is used to configure features pertinent to the FIFO mode of operation. This is write-only.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 2h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Receiver Trigger Level (RTL):** Trigger level in bytes for the Receiver FIFO. Once the trigger level number of bytes is reached, an interrupt will be sent to the Host [7:6] RCVR FIFO Trigger Level (bytes)<br>• 00 -- 01<br>• 01 -- 04<br>• 10 -- 08<br>• 11 -- 14 |
| 5:4 | 0x0<br>WO | **Reserved (RSVD):** Writes to this field shall be dropped. |
| 3 | 0x0<br>WO | **RDY Mode (RDYM):** This bit has no effect on HW performance. |
| 2 | 0x0<br>WO | **XMT FIFO Clear (XFIC):** When the Host writes one to this bit the HW will clear the XMT FIFO. This bit is self-cleared by HW. |
| 1 | 0x0<br>WO | **RCV FIFO Clear (RFIC):** When the Host writes one to this bit the HW will clear the RCV FIFO. This bit is self-cleared by HW. |
| 0 | 0x0<br>WO | **FIFO Enable (FIE):** When set indicates that the KT interface is working in FIFO node. When this bit value is changed the RCV and XMT FIFO are cleared by HW. This bit must be a 1 when other FCR bits are written to or they will NOT be programmed. |

## 52.12.8 KT Line Control register (IO_KTLCR)—Offset 3h

This register implements the standard Line Control register in the KT Block. It specifies the format of the asynchronous data communications exchange and sets the DLAB bit. Most bits in this register have no effect on HW and are only used by the FW.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 03h

**BAR and Offset:** [KT_HOST_IOBAR] + 3h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Divisor Latch Access Bit (DLAB):**<br>• 1: Allows access to the Divisor Latch Registers and reading of the FIFO Control Register.<br>• 0: Allows access to RBR, THR, IER and IIR registers. |
| 6 | 0x0<br>RW | **Break Control (BC):** This bit has no effect on HW. |
| 5:4 | 0x0<br>RW | **Parity Bit Mode (PBM):** This bit has no effect on HW. |
| 3 | 0x0<br>RW | **Parity Enable (PE):** This bit has no effect on HW. |
| 2 | 0x0<br>RW | **Stop Bit Select (SBS):** This bit has no effect on HW. |
| 1:0 | 0x3<br>RW | **Word Select Byte (WSB):** This bit has no effect on HW. |

## 52.12.9 KT Modem Control register (IO_KTMCR)—Offset 4h

This register implements the standard Modem Control register in the KT Block. It controls the interface with the modem. Because the FW emulates the modem, the Host communicates with the FW via this register. This register has impact on the HW in the loopback mode.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 4h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4 | 0x0<br>RW | **Loop Back Mode (LBM):** When set by Host indicates that the serial port is in loop Back mode. This means that the data that is transmitted by the host should be received.<br>Helps in debug of the interface |
| 3 | 0x0<br>RW | **Output 2 (OUT2):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit (7) |
| 2 | 0x0<br>RW | **Output 1 (OUT1):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit(6) |
| 1 | 0x0<br>RW | **Request to Send Out (RTSO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value of this bit is written by HW to Modem Status Register bit(4) |
| 0 | 0x0<br>RW | **Data Terminal Ready Out (DRTO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value in this bit is written by HW to Modem Status Register Bit(5) |

## 52.12.10 KT Line Status register (IO_KTLSR)—Offset 5h

This register implements the standard Line Status register in the KT Block. It provides status information of the data transfer to the Host. Error indication etc are provided by the HW/FW to the host via this register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 5h

**Default:** 60h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/C/V | **RCVR FIFO Error (RXFER):** This bit is cleared in non FIFO mode.<br>This bit is connected to UART LSR[BI] bit in FIFO mode. |
| 6 | 0x1<br>RO/C/V | **Transmitter Empty (TEMT):** In non-FIFO mode:<br>this bit will be set to 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, OR cleared to 0 whenever either THR or TSR contains a data byte.<br><br>In FIFO mode:<br>this bit will be set to 1 whenever XMIT FIFO and TSR are both empty, OR cleared to 0 whenever either XMIT FIFO or TSR contains a data byte.<br><br>Implementation note: This bit is connected by HW to THRE bit. |
| 5 | 0x1<br>RO/C/V | **Transmitter Holding Register Empty (THRE):** The bit will always be set whenever Host changes UART FCR[FIE] bit.<br>This bit is controlled differently in FIFO vs non-FIFO modes:<br>Non-FIFO mode: This bit will be cleared by HW when Host writes to THR, and set by HW when FW reads from THR.<br>FIFO mode: This bit will be set by HW when XMIT FIFO is empty, and cleared by HW when XMIT FIFO is NOT empty. |
| 4 | 0x0<br>RO/C/V | **Break Interrupt (BI):** This bit will be cleared by HW when the UART LSR register is read by Host.<br>This bit will be set by HW in two cases:<br>In FIFO mode, when FW sets the KTRIVR[SBI] bit.<br>In non-FIFO mode, when FW sets the KTRBR[BIA] bit. |
| 3 | 0x0<br>RO | **Framing Error (FE):** This bit is NOT implemented and hardwired to 0. |
| 2 | 0x0<br>RO | **Parity Error (PE):** This bit is NOT implemented and hardwired to 0. |
| 1 | 0x0<br>RO/C/V | **Overrun Error (OE):** HW will clear this bit when Host reads from KTLSR register and this is source of interrupt.<br>HW will set this bit when overrun error is detected.<br>In non-FIFO mode, the character that triggers the overrun error shall overwrite the character in the RBR register.<br>In FIFO mode, the character that triggers the overrun error shall be dropped without corrupting the RCVR FIFO. |
| 0 | 0x0<br>RO/V | **Data Ready (DR):** Non-FIFO Mode: this bit will be set by HW when FW writes to RBR and cleared by HW when Host reads from RBR.<br>FIFO Mode: the bit will be set by HW when RCVR FIFO is not empty and cleared by HW when RCVR FIFO is empty. |

## 52.12.11 KT Modem Status register (IO_KTMSR)—Offset 6h

This register implements the standard Modem Status register in the KT Block. It provides the current state of the control lines from the modem. The modem functionality is emulated by the FW.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 6h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Data Carrier Detect (DCD):** In Loop Back mode this bit is connected by HW to the value of MCR bit 3 |
| 6 | 0x0<br>RO | **Ring Indicator (RI):** In Loop Back mode this bit is connected by HW to the value of MCR bit 2. |
| 5 | 0x0<br>RO | **Data Set Ready (DSR):** In Loop Back mode this bit is connected by HW to the value of MCR bit 0. |
| 4 | 0x0<br>RO | **Clear To Send (CTS):** In Loop Back mode this bit is connected by HW to the value of MCR bit 1. |
| 3 | 0x0<br>RO/C/V | **Delta Data Carrier Detect (DDCD):** This bit is set when bit 7 is changed.<br>This bit is cleared by HW when the MSR register is being read by the HOST driver. |
| 2 | 0x0<br>RO/C/V | **Trailing Edge of Read Detector (TERI):** This bit is set when bit 6 is changed from 1 to 0.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 1 | 0x0<br>RO/C/V | **Delta Data Set Ready (DDSR):** This bit is set when bit 5 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 0 | 0x0<br>RO/C/V | **Delta Clear To Send (DCTS):** This bit is set when bit 4 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |

## 52.12.12 KT Scratch register (IO_KTSCR)—Offset 7h

This register implements the standard Scratch register in the KT Block. It allows Host SW programmers to hold temporary data. It has NO effect on HW.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 7h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:24, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Scratch Register Data (SCRD):** Reserved. |

## 52.13 Registers in I/O Space—IDE_HOST_PCMDIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_PCMDIOBAR described in Section 52.6.5, "IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h" on page 2308.

### 52.13.1 IDE Data Register (IDEDATA)—Offset 0h

This register serves as the data interface in IDE PIO modes. Unit of access is in WORD. Host writes to this register are converted into writes to the PIO buffer in CSxE memory at monotonically increasing WORD address. Host writes are dropped under any one of the following conditions: (1) PIO modes are disabled. (Cfg. IDEPBPR[EN] defined in CSxE memory space) (2) PIO buffer is full. Host reads from this register are converted into reads from the PIO buffer in CSxE memory at monotonically increasing WORD address. Host reads are dropped under any one of the following conditions: (1) PIO modes are disabled. (Cfg. IDEPBPR[EN] defined in CSxE memory space) (2) PIO buffer is empty.

Reset condition: no connection to any reset signal.

**Type:** I/O Register
(Size: 16 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 0h

**Default:** 0000h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/V | **IDE Data Register (IDEDR):** Data interface in IDE PIO modes. Host reads/writes from/to this register are converted into reads/writes from/to the PIO buffer in CSxE memory. |

### 52.13.2 IDE Features Register (IDEFR)—Offset 1h

This register implements the standard Features register in the Primary IDE Command Block. This is write-only. BE[0] shall be deasserted.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Features Data (IDEFD):** The IDE Features register is command specific and is used to enable and disable interface features. |

## 52.13.3 IDE Error Register DEV0 (IDEERD0)—Offset 1h

This register implements the standard Device 0 Error register in the Primary IDE Command Block. This is read-only. BE[0] shall be deasserted. DEV = 0 (i.e. IDEDHIR[4] = 0).

Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Error Data (IDEED):** Upon command completion of any command except EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET, the content of this register is valid and contains error status when the ERR bit is set to 1 in the Status register (i.e. IDESR0[ERR] = 1).<br>Following a power-on, a hardware or software reset, or command completion of an EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET command, this register contains a diagnostic code.<br><br>Bit 2 ABRT (command aborted) is set to 1 to indicate the requested command has been command aborted because the command code or a command parameter is invalid, the command is not supported, a pre-requisite for the command has not been met, or some other error has occurred. |

## 52.13.4 IDE Error Register DEV1 (IDEERD1)—Offset 1h

This register implements the standard Device 1 Error register in the Primary IDE Command Block. This is read-only. BE[0] shall be deasserted. DEV = 1 (i.e. IDEDHIR[4] = 1).

Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Error Data (IDEED):** Upon command completion of any command except EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET, the content of this register is valid and contains error status when the ERR bit is set to 1 in the Status register (i.e. IDESR0[ERR] = 1).<br>Following a power-on, a hardware or software reset, or command completion of an EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET command, this register contains a diagnostic code.<br><br>Bit 2 ABRT (command aborted) is set to 1 to indicate the requested command has been command aborted because the command code or a command parameter is invalid, the command is not supported, a pre-requisite for the command has not been met, or some other error has occurred. |

## 52.13.5 IDE Sector Count In Register (IDESCIR)—Offset 2h

This register implements the standard Sector Count In register in the Primary IDE Command Block. All 3 registers (IDESCIR, IDESCOR0, IDESCOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is meaningful for an ATA command while unused for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Sector Count Data (IDESCD):** The Sector Count register specifies the sector count of the data for which an ATA transfer has been made. |

## 52.13.6 IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h

This register implements the standard Device 0 Sector Count Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA/ATAPI command. This register is called IDE Interrupt Reason Register DEV0 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Count Out DEV0 (ISCOD0):** Sector Count Out register for Device 0 |

## 52.13.7 IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h

This register implements the standard Device 1 Sector Count Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA/ATAPI command. This register is called IDE Interrupt Reason Register DEV1 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Count Out DEV1 (ISCOD1):** Sector Count Out register for Device 1 |

## 52.13.8 IDE Sector Number In Register (IDESNIR)—Offset 3h

This register implements the standard Sector Number In register in the Primary IDE Command Block in CHS addressing mode. All 3 registers (IDESNIR, IDESNOR0, IDESNOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA Low In Register in LBA addressing mode. This register is reserved for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Sector Number Data (IDESND):** The Sector Number register is set by the host to specify the starting sector number associated with the next ATA command to be executed.<br>Following a qualified ATA command sequence the device will set the register value to the last sector read or written as a result of the previous ATA command. |

## 52.13.9 IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h

This register implements the standard Device 0 Sector Number Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Low Out Register DEV0 in LBA addressing mode. This register is reserved for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Number Out DEV0 (IDESNO0):** Sector Number Out register for Device 0 |

## 52.13.10 IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h

This register implements the standard Device 1 Sector Number Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Low Out Register DEV1 in LBA addressing mode. This register is reserved for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Number Out DEV1 (IDESNO1):** Sector Number Out register for Device 1 |

## 52.13.11 IDE Cylinder Low In Register (IDECLIR)—Offset 4h

This register implements the standard Cylinder Low In register in the Primary IDE Command Block. All 3 registers (IDECLIR, IDCLOR0, IDCLOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA Mid In Register in LBA addressing mode. This register is called IDE Byte Count Low Limit Register for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Cylinder Low Data. (IDECLD):** The Cylinder Low register is set by the host to specify the cylinder number low byte.<br>Following an ATA command, the contents of this register is written by the device, identifying the cylinder number low byte. |

## 52.13.12 IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h

This register implements the standard Device 0 Cylinder Low Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Mid Out Register DEV0 in LBA addressing mode. This register is called IDE Byte Count Low Register DEV0 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder Low Out DEV0. (IDECLO0):** Cylinder Low Out Register for Device 0. |

## 52.13.13 IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h

This register implements the standard Device 1 Cylinder Low Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Mid Out Register DEV1 in LBA addressing mode. This register is called IDE Byte Count Low Register DEV1 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder Low Out DEV1. (IDECLO1):** Cylinder Low Out Register for Device 1. |

## 52.13.14 IDE Cylinder High In Register (IDECHIR)—Offset 5h

This register implements the standard Cylinder High In register in the Primary IDE Command Block. All 3 registers (IDECHIR, IDCHOR0, IDCHOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA High In Register in LBA addressing mode. This register is called IDE Byte Count High Limit Register for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Cylinder High Data (IDECHD):** The Cylinder High register is set by the host to specify the cylinder number high byte. Following an ATA command, the contents of this register is written by the device, identifying the cylinder number high byte. |

## 52.13.15 IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h

This register implements the standard Device 0 Cylinder High Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA High Out Register DEV0 in LBA addressing mode. This register is called IDE Byte Count High Register DEV0 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder High Out DEV0 (IDECHO0):** Cylinder High Out register for Device 0 |

## 52.13.16 IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h

This register implements the standard Device 1 Cylinder High Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA High Out Register DEV1 in LBA addressing mode. This register is called IDE Byte Count High Register DEV1 for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder High Out DEV1 (IDECHO1):** Cylinder High Out register for Device 1 |

## 52.13.17 IDE Drive/Head In Register (IDEDHIR)—Offset 6h

This register implements the standard Drive/Head In register in the Primary IDE Command Block. All 3 registers (IDEDHIR, IDDHOR0, IDDHOR1) will be updated with the same write data when this register is write-accessed. This is write-only. Bit 4 of this register is the DEV (master/slave) bit. This bit is cleared by hardware on IDE software reset (IDDCR[SRST] = 1) in addition to Host reset. This register is called IDE Device In Register in LBA addressing mode or for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Drive/Head Data (IDEDHD):** The Drive/Head Register is used by the host and the device to select the type of addressing (CHS or LBA), the drive letter, and either bits 3 through 0 of the head number in CHS mode or in LBA mode, bits D3:D0 reflect the logical block number bits LBA27:LBA24. |

## 52.13.18 IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h

This register implements the standard Device 0 Drive/Head Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE Device Out Register DEV0 in LBA addressing mode or for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Drive/Head Out DEV0 (IDEDHO0):** Drive/Head Out register of Device 0 |

## 52.13.19 IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h

This register implements the standard Device 1 Drive/Head Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE Device Out Register DEV1 in LBA addressing mode or for an ATAPI command.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Drive/Head Out DEV1 (IDEDHO1):** Drive/Head Out register of Device 1 |

## 52.13.20 IDE Command Register (IDECR)—Offset 7h

This register implements the standard Command register in the Primary IDE Command Block. This is write-only.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Command Data (IDECD):** The Command register specifies the ATA/ATAPI Command code being issued to the drive by the host.<br>Command processing begins when this register is written. The content of the Command Block registers becomes parameters of the command when this register is written.<br>Writing this register clears any pending interrupt condition.<br>Please refer to the description of the INTM and INTS bits in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. |

## 52.13.21 IDE Status Register DEV0 (IDESR0)—Offset 7h

This register implements the standard Device 0 Status register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA/ATAPI command. Additionally upon completion of the last data phase of a PIO transfer, HW will load this register with value of the SRAFP field in the IDESRAP register in the CSxE memory space. Host read of this register clears Device 0's interrupt. Please refer to the description of the INTM bit in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. The bit description is for ATA mode. Different name in ATAPI mode, if applicable, is cited as well.

Reset condition: CSxE reset with the BSY bit as an exception. Please refer to description of the BSY bit below for a list of BSY bit reset conditions. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**Default:** 80h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Busy (BSY):** This bit is set by HW when the IDECR is being written and DEV=0, OR when IDDCR[SRST] bit is asserted by Host, OR host system reset, OR D3 to D0 transition of the IDE function.<br>This bit is cleared by FW write of 0.<br>Please refer to general register description above for special handling for a PIO transfer.<br>When this bit is 1, any other bit in this register is meaningless. |
| 6 | 0x0<br>RO/V | **Drive Ready (DRDY):** When set, indicates drive is ready for command.<br>Update of this bit is completely FW-assisted.<br>This bit is also called Device Ready.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 5 | 0x0<br>RO/V | **Drive Fault (DF):** When set, indicates Error on the drive.<br>Update of this bit is completely FW-assisted.<br>This bit is called DMA Ready for a DMA ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 4 | 0x0<br>RO/V | **Drive Seek Complete (DSC):** When set, indicates Heads are positioned over the desired cylinder.<br>Update of this bit is completely FW-assisted.<br>This bit is called Service for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 3 | 0x0<br>RO/V | **Data Request (DRQ):** Set when the drive wants to exchange data with the Host via the data register.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 2 | 0x0<br>RO/V | **Corrected Data (CORR):** When set, indicates a correctable read error has occurred.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 1 | 0x0<br>RO/V | **Index (IDX):** This bit is set once per rotation of the medium when the index mark passes under the read/write head.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 0 | 0x0<br>RO/V | **Error (ERR):** When set, indicates an error occurred in the process of executing the current command. The Error Register of the selected device contains the error information.<br>Update of this bit is completely FW-assisted.<br>This bit is called Check for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |

## 52.13.22 IDE Status Register DEV1 (IDESR1)—Offset 7h

This register implements the standard Device 1 Status register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA/ATAPI command. Additionally upon completion of the last data phase of a PIO transfer, HW will load this register with value of the SRAFP field in the IDESRAP register in the CSxE memory space. Host read of this register clears Device 1's interrupt. Please refer to the description of the INTM bit in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. The bit description is for ATA mode. Different name in ATAPI mode, if applicable, is cited as well.

Reset condition: CSxE reset with the BSY bit as an exception. Please refer to description of the BSY bit below for a list of BSY bit reset conditions. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**Default:** 80h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:24, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Busy (BSY):** This bit is set by HW when the IDECR is being written and DEV=1, OR when IDDCR[SRST] bit is asserted by Host, OR host system reset, OR D3 to D0 transition of the IDE function.<br>This bit is cleared by FW write of 0.<br>Please refer to general register description above for special handling for a PIO transfer.<br>When this bit is 1, any other bit in this register is meaningless. |
| 6 | 0x0<br>RO/V | **Drive Ready (DRDY):** When set, indicates drive is ready for command.<br>Update of this bit is completely FW-assisted.<br>This bit is also called Device Ready.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 5 | 0x0<br>RO/V | **Drive Fault (DF):** When set, indicates Error on the drive.<br>Update of this bit is completely FW-assisted.<br>This bit is called DMA Ready for a DMA ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 4 | 0x0<br>RO/V | **Drive Seek Complete (DSC):** When set, indicates Heads are positioned over the desired cylinder.<br>Update of this bit is completely FW-assisted.<br>This bit is called Service for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 3 | 0x0<br>RO/V | **Data Request (DRQ):** Set when the drive wants to exchange data with the Host via the data register.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 2 | 0x0<br>RO/V | **Corrected Data (CORR):** When set, indicates a correctable read error has occurred.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 1 | 0x0<br>RO/V | **Index (IDX):** This bit is set once per rotation of the medium when the index mark passes under the read/write head.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 0 | 0x0<br>RO/V | **Error (ERR):** When set, indicates an error occurred in the process of executing the current command. The Error Register of the selected device contains the error information.<br>Update of this bit is completely FW-assisted.<br>This bit is called Check for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |

## 52.14 Registers in I/O Space—IDE_HOST_PCTLIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_PCTLIOBAR described in described in Section 52.6.6, "IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h" on page 2308.

### 52.14.1 IDE Device Control Register (IDDCR)—Offset 2h

This register implements the standard Device Control register in the Primary IDE Control Block. This is write-only. BE[0] shall be deasserted.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCTLIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCTLIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCTLIOBAR Reference:** [B:0, D:24, F:2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **Reserved (RSVD1):** Writable by Host, but NO HW effect due to writes. |
| 2 | 0x0<br>WO | **Software Reset (SRST):**<br>• 1: both devices shall perform the software reset protocol.<br>• 0: both devices shall exit software reset. |
| 1 | 0x0<br>WO | **Host Interrupt Disable (nIEN):**<br>• 1: disables HW from generating interrupt to the Host.<br>• 0: enables HW to generate interrupt to the Host. |
| 0 | 0x0<br>WO | **Reserved (RSVD0):** Writable by Host, but NO HW effect due to writes. |

### 52.14.2 IDE Alternate Status Register (IDASR)—Offset 2h

This register implements the standard Alternate Status register in the Primary IDE Control Block. This is read-only. It shadows the standard Status registers in the Primary IDE Command Block. Read of this register does NOT clear the interrupt status bits in the selected Status register. IDESR0 is shadowed when DEV = 0 (i.e. IDEDHIR[4] = 0). IDESR1 is shadowed when DEV = 1 (i.e. IDEDHIR[4] = 1).

Reset condition: There is no reset associated with this register because it is NOT a physical register.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCTLIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCTLIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCTLIOBAR Reference:** [B:0, D:24, F:2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Alternate Status Register (IDEASR):** Shadows DEV0/DEV1 Status register on Host reads depending on IDEDHIR[4] setting. |

## 52.15 Registers in I/O Space—IDE_HOST_BMIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_BMIOBAR described in described in Section 52.6.9, "IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h" on page 2310.

### 52.15.1 IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h

This register implements the standard IDE Primary Bus Master Command register in the IDE Bus Master Block.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 0h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** Not implemented. Hardwired to 0. |
| 3 | 0x0<br>RW | **Read or Write Control (RWC):** This bit sets the direction of the bus master transfer.<br>• 0: reads are performed from Host system memory.<br>• 1: writes are performed to Host system memory.<br>This bit shall NOT be changed when the bus master function is active. |
| 2:1 | 0x0<br>RO | **Reserved (RSVD0):** Not implemented. Hardwired to 0. |
| 0 | 0x0<br>RW/V | **Start/Stop Bus Master (SSBM):** Host starts a bus master operation in the device by setting this bit to 1. The device shall only transfer data between the ATA device and Host memory in the bus master operation.<br>Host stops the bus master operation by clearing this bit to 0. All state information will be lost when 0 is written.<br><br>Host CANNOT stop and then resume the bus master operation.<br>When Host clears this bit to 0 while the bus master operation is still active, the bus master operation will be aborted and data transferred from the device may be discarded before being written to host memory.<br>In any event the device should terminate any DMA burst in progress compliant with the ATA protocol being used.<br>If the UDMA protocol is being used, there may be a CRC error report. The ATA device may be in an indeterminate state.<br><br>HW will clear this bit to 0 after the data transfer has been completed, as indicated by satisfaction of either of the following conditions:<br>(1) IDEPBMSR[BMIA] is cleared to 0. OR<br>(2) IDEPBMSR[INT] is set to 1. |

## 52.15.2 IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h

This register implements the standard IDE Primary Bus Master Device Specific 0 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data.

Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 1h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Device Specific Data0 (DSD0):** Device Specific |

## 52.15.3 IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h

This register implements the standard IDE Primary Bus Master Status register in the IDE Bus Master Block.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 2h

**Default:** 80h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Simplex Only (SO):** This bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time.<br>If the bit is 0, both channels operate independently and can be used at the same time.<br>If the bit is 1, only one channel may be used at a time.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 6 | 0x0<br>RW/V | **Device 1 DMA Capable (D1DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 1 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance.<br><br>Reset condition: Host reset.<br>Non-sticky. |
| 5 | 0x0<br>RW/V | **Device 0 DMA Capable (D0DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 0 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance.<br><br>Reset condition: Host reset.<br>Non-sticky. |
| 4:3 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RW/1C/V | **Interrupt (INT):** HW shall set this bit to 1 on rising edge detection of the ATA channels interrupt line (INTRQ).<br>Host SW writes 1 to clear this bit. Host SW can use this bit to determine if an ATA device has asserted its interrupt line (INTRQ) (see ATA standard).<br>When this bit is 1, all data MUST have been transferred to the Hosts system memory.<br>The adapter shall not set this bit to 1 until it has flushed all internal data buffers.<br><br>NOTE: When INTRQ is deasserted, HW won't clear this bit atomically.<br>When INTRQ is deasserted, this bit shall be cleared as well by Host SW writing 1 to this bit.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 1 | 0x0<br>RW/1C/V | **Error (ER):** FW will set this bit to 1 when the device encounters an error in transferring data to/from Host memory.<br>The exact error condition is bus specific and can be determined in a bus specific manner.<br>Host SW writes 1 to clear this bit.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **Bus Master IDE Active (BMIA):** HW shall set this bit when<br>(1) Host sets the IDEPBMCR[SSBM] bit.<br><br>FW shall clear this bit when<br>(1) The last transfer for a region is performed, where EOT for that region is set in the region descriptor. OR<br>(2) FW detects that Host cleared IDEPBMCR[SSBM] bit as indicated by a value of 1 in proxy bit IDEBMCR[SSBMC] in CSxE root space.<br><br>Before clearing this bit to 0, FW must satisfy the following conditions in addition:<br>(1) All data transferred from the device during the previous bus master command has been transferred to the Hosts system memory, unless the bus master command was aborted. OR<br>(2) All data has been written to the device.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |

## 52.15.4 IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h

This register implements the standard IDE Primary Bus Master Device Specific 1 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data.

Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 3h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Device Specific Data1 (DSD1):** Device Specific Data |

## 52.15.5 IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h

This register implements the Byte 0 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 4h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 0 (DTPB0):** Byte 0 of the Descriptor Table pointer.<br>The Descriptor Table shall be DW-aligned.<br>Hence bit[1:0] shall always be 0. |

## 52.15.6 IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h

This register implements the Byte 1 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 5h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 1 (DTPB1):** Byte 1 of the Descriptor Table pointer. |

## 52.15.7 IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h

This register implements the Byte 2 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 6h

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 2 (DTPB2):** Byte 2 of the Descriptor Table pointer. |

## 52.15.8 IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h

This register implements the Byte 3 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 7h

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 3 (DTPB3):** Byte 3 of the Descriptor Table pointer. |

## 52.15.9 IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h

This register implements the standard IDE Secondary Bus Master Command register in the IDE Bus Master Block. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 8h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** Not implemented.<br>Hardwired to 0. |
| 3 | 0x0<br>RW | **Read or Write Control (RWC):** This bit sets the direction of the bus master transfer.<br>• 0: reads are performed from Host system memory.<br>• 1: writes are performed to Host system memory.<br>This bit shall NOT be changed when the bus master function is active. |
| 2:1 | 0x0<br>RO | **Reserved (RSVD0):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RW | **Start/Stop Bus Master (SSBM):** Host starts a bus master operation in the device by setting this bit to 1.<br>The device shall only transfer data between the ATA device and Host memory in the bus master operation.<br>Host stops the bus master operation by clearing this bit to 0. All state information will be lost when 0 is written.<br><br>Host CANNOT stop and then resume the bus master operation. |

## 52.15.10 IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h

This register implements the standard IDE Secondary Bus Master Device Specific 0 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 9h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Device Specific Data0 (DSD0):** Device Specific |

## 52.15.11 IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah

This register implements the standard IDE Secondary Bus Master Status register in the IDE Bus Master Block. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset with the SO bit as an exception. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Ah

**Default:** 80h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Simplex Only (SO):** Value indicates whether both Bus Master Channels can be operated at the same time or not.<br>• 0: Both can be operated independently<br>• 1: Only one can be operated at a time.<br><br>This bit shadows IDEPBMSR[SO].<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 6 | 0x0<br>RW | **Device 1 DMA Capable (D1DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 1 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance. |
| 5 | 0x0<br>RW | **Device 0 DMA Capable (D0DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 0 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance. |
| 4:3 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RO | **Interrupt (INT):** No functionality implemented.<br>Hardwired to 0. |
| 1 | 0x0<br>RO | **Error (ER):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RO | **Bus Master IDE Active (BMIA):** Not implemented.<br>Hardwired to 0. |

## 52.15.12 IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh

This register implements the standard IDE Secondary Bus Master Device Specific 1 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Bh

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Device Specific Data1 (DSD1):** Device Specific Data |

## 52.15.13 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch

This register implements the Byte 0 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Ch

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 0 (DTPB0):** Byte 0 of the Descriptor Table pointer.<br>The Descriptor Table shall be DW-aligned. Hence bit[1:0] shall always be 0. |

## 52.15.14 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh

This register implements the Byte 1 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Dh

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 1 (DTPB1):** Byte 1 of the Descriptor Table pointer. |

## 52.15.15 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh

This register implements the Byte 2 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Eh

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 2 (DTPB2):** Byte 2 of the Descriptor Table pointer. |

## 52.15.16 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh

This register implements the Byte 3 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register.

Reset condition: Host reset. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Fh

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:24, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 3 (DTPB3):** Byte 3 of the Descriptor Table pointer. |

## 52.16 Sideband Registers—KT (Host)

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 52.16.1 KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h

This register provides for BIOS to specify which interrupt pin KT shall use in the Host root space when generating an interrupt in the PCI legacy mode.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xe4] + 0h
**Host Memory Space:** SBREG_BAR + 0xe40000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 7:0 | 0x0<br>RW/O | **Interrupt Pin (IPIN):** BIOS writes to this field once to specify which interrupt pin KT shall use. After written once, this register will become read-only.<br>This field shall be shadowed into the Interrupt Pin register in the PCI header of the KT Host PCI function.<br><br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved<br><br>Reset condition: Host side reset.<br>Non-sticky. |

### 52.16.2 BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h

This register provides for BIOS to enable or disable the KT Host PCI function.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xe4] + 4h
**Host Memory Space:** SBREG_BAR + 0xe40000 + 4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RW | **Host PCI Function Disable (HDIS):** BIOS sets this bit to disable the KT Host PCI function.<br>BIOS clears this bit to enable the KT Host PCI function.<br><br>Reset condition: Host side reset.<br>Non-sticky. |

## 52.17 Sideband Registers—IDE (Host)

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 52.17.1 IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h

This register provides for BIOS to specify which interrupt pin IDE shall use in the Host root space when generating an interrupt in the PCI legacy mode.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xe4] + 0h
**Host Memory Space:** SBREG_BAR + 0xe40000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Not implemented. Hardwired to 0. |
| 7:0 | 0x0<br>RW/O | **Interrupt Pin (IPIN):** BIOS writes to this field once to specify which interrupt pin IDE shall use. After written once, this register will become read-only.<br>This field shall be shadowed into the Interrupt Pin register in the PCI header of the IDE Host PCI function.<br><br>Value Decoding<br>00h The function does NOT use an interrupt pin.<br>01h INTA<br>02h INTB<br>03h INTC<br>04h INTD<br>05h - FFh Reserved<br><br>Reset condition: Host side reset.<br>Non-sticky. |

### 52.17.2 BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h

This register provides for BIOS to enable or disable the IDE Host PCI function.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xe4] + 4h
**Host Memory Space:** SBREG_BAR + 0xe40000 + 4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Not implemented. Hardwired to 0. |
| 0 | 0x0<br>RW | **Host PCI Function Disable (HDIS):** BIOS sets this bit to disable the IDE Host PCI function. BIOS clears this bit to enable the IDE Host PCI function.<br><br>Reset condition: Host side reset.<br>Non-sticky. |

§ §

# 53 HSUART Controller - B0, D26, F(0, 1, 2)

## 53.1 Introduction and Index

The host-accessible registers for the HSUART Controller are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, “Features by product SKU - Server and Cloud Storage SKUs 0 through 4”
- Table 1-2, “Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10”
- Table 1-3, “Features by Product SKU - Extended Temperature SKUs 11 through 14”.

The legend for the register access codes are described in Chapter 43, “Introduction.”

### 53.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 26 (decimal), Function (0, 1, 2). The offset addresses are listed.

**Table 53-1. Summary of PCI Configuration Registers—0/26/0,1,2**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19D88086 | "VENDOR_DEVICE_ID—Offset 0h" on page 2387 |
| 4 | 4 | 00100000 | "CMD_STAT_REG—Offset 4h" on page 2388 |
| 8 | 4 | 07000200 | "REV_CLASS_CODE—Offset 8h" on page 2390 |
| C | 4 | 00800000 | "CACHE—Offset Ch" on page 2390 |
| 10 | 4 | 00000001 | "IOBA—Offset 10h" on page 2391 |
| 14 | 4 | 00000000 | "MEMBA—Offset 14h" on page 2391 |
| 2C | 4 | 00008086 | "SUBSYSTEM_VENDOR_ID—Offset 2Ch" on page 2392 |
| 34 | 4 | 00000040 | "CAP_PTR—Offset 34h" on page 2392 |
| 3C | 4 | 000001FF | "INTERRUPT—Offset 3Ch" on page 2392 |
| 40 | 4 | 00005005 | "MSIC—Offset 40h" on page 2393 |
| 44 | 4 | 00000000 | "MSIA—Offset 44h" on page 2394 |
| 48 | 4 | 00000000 | "MSID—Offset 48h" on page 2394 |
| 50 | 4 | 00030001 | "PMC—Offset 50h" on page 2395 |
| 54 | 4 | 00000008 | "PMCSR—Offset 54h" on page 2396 |
| 60 | 4 | 00000000h | "DSC—Offset 60h" on page 2397 |
| 74 | 4 | 00000000h | "FUNC_RDCFG_HIDE—Offset 74h" on page 2398 |

### 53.1.2 Host Memory Space—MEMBA

An index of the registers that are accessible in Host Memory Space is shown here. There is a set of these registers for each of the three HSUART Functions (F0, F1, and F2).

**Table 53-2. Summary of Memory Mapped I/O Registers—MEMBA**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "RBR_THR_DLL—Offset 0h" on page 2399 |
| 1 | 1 | 00 | "IER_DLH—Offset 1h" on page 2399 |
| 2 | 1 | 00 | "IIR_FCR—Offset 2h" on page 2400 |
| 3 | 1 | 00 | "LCR—Offset 3h" on page 2400 |
| 4 | 1 | 00 | "MCR—Offset 4h" on page 2402 |
| 5 | 1 | 60 | "LSR—Offset 5h" on page 2404 |
| 6 | 1 | 00 | "MSR—Offset 6h" on page 2406 |
| 7 | 1 | 00 | "SPR—Offset 7h" on page 2408 |
| 8 | 4 | 00000000 | "UART_FISR—Offset 8h" on page 2408 |
| 20 | 4 | 00000000 | "RFOR—Offset 20h" on page 2408 |
| 24 | 4 | 00000010 | "ABR—Offset 24h" on page 2409 |
| 28 | 4 | 00000000 | "ACR—Offset 28h" on page 2410 |
| 30 | 4 | 00000010 | "PSR—Offset 30h" on page 2410 |
| 34 | 4 | 000000D8 | "UCMR—Offset 34h" on page 2410 |
| 38 | 4 | 00003D09 | "UCDR—Offset 38h" on page 2411 |
| 80 | 4 | 000F0000 | "CH0SR—Offset 80h" on page 2412 |
| 84 | 4 | 00000000 | "CH0CR—Offset 84h" on page 2412 |
| 88 | 4 | 00008000 | "CH0DCR—Offset 88h" on page 2414 |
| 90 | 4 | 00000000 | "CH0BSR—Offset 90h" on page 2415 |
| 94 | 4 | 00000000 | "CH0MTSR—Offset 94h" on page 2415 |
| A0 | 4 | 00000000 | "CH0D0SAR—Offset A0h" on page 2415 |
| A4 | 4 | 00000000 | "CH0D0TSR—Offset A4h" on page 2416 |
| A8 | 4 | 00000000 | "CH0D1SAR—Offset A8h" on page 2416 |
| AC | 4 | 00000000 | "CH0D1TSR—Offset ACh" on page 2416 |
| B0 | 4 | 00000000 | "CH0D2SAR—Offset B0h" on page 2417 |
| B4 | 4 | 00000000 | "CH0D2TSR—Offset B4h" on page 2417 |
| B8 | 4 | 00000000 | "CH0D3SAR—Offset B8h" on page 2418 |
| BC | 4 | 00000000 | "CH0D3TSR—Offset BCh" on page 2418 |
| C0 | 4 | 000F0000 | "CH1SR—Offset C0h" on page 2419 |
| C4 | 4 | 00000000 | "CH1CR—Offset C4h" on page 2420 |
| C8 | 4 | 00008000 | "CH1DCR—Offset C8h" on page 2421 |
| D0 | 4 | 00000000 | "CH1BSR—Offset D0h" on page 2422 |
| D4 | 4 | 00000000 | "CH1MTSR—Offset D4h" on page 2422 |
| E0 | 4 | 00000000 | "CH1D0SAR—Offset E0h" on page 2422 |
| E4 | 4 | 00000000 | "CH1D0TSR—Offset E4h" on page 2423 |

## Table 53-2. Summary of Memory Mapped I/O Registers—MEMBA (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| E8 | 4 | 00000000 | "CH1D1SAR—Offset E8h" on page 2423 |
| EC | 4 | 00000000 | "CH1D1TSR—Offset ECh" on page 2423 |
| F0 | 4 | 00000000 | "CH1D2SAR—Offset F0h" on page 2424 |
| F4 | 4 | 00000000 | "CH1D2TSR—Offset F4h" on page 2424 |
| F8 | 4 | 00000000 | "CH1D3SAR—Offset F8h" on page 2425 |
| FC | 4 | 00000000 | "CH1D3TSR—Offset FCh" on page 2425 |

### 53.1.3 Host I/O Space—IOBA

An index of the registers that are accessible in Host I/O Space is shown here. There is a set of these registers for each of the three HSUART Functions (F0, F1, and F2).

**Table 53-3. Summary of I/O Registers—IOBA**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "RBR_THR_DLL—Offset 0h" on page 2426 |
| 1 | 1 | 00 | "IER_DLH—Offset 1h" on page 2426 |
| 2 | 1 | 00 | "IIR_FCR—Offset 2h" on page 2427 |
| 3 | 1 | 00 | "LCR—Offset 3h" on page 2428 |
| 4 | 1 | 00 | "MCR—Offset 4h" on page 2430 |
| 5 | 1 | 00 | "LSR—Offset 5h" on page 2432 |
| 6 | 1 | 00 | "MSR—Offset 6h" on page 2434 |
| 7 | 1 | 00 | "SPR—Offset 7h" on page 2436 |

## 53.2 Registers in Configuration Space

### 53.2.1 VENDOR_DEVICE_ID—Offset 0h

*Note:* The default value of DEVICE_ID can be overwritten using the setidval sideband message.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 0h

**Default:** 19D88086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19d8<br>RO/V | **DEVICE_ID:** Device ID |
| 15:0 | 0x8086<br>RO | **VENDOR_ID:** Identifies uniquely Intel products. |

## 53.2.2 CMD_STAT_REG—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 4h

**Default:** 00100000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **STATUS_DETECTED_PARITY_ERROR:** Detected Parity Error.<br>This bit is set when the device receives an uncorrectable data error or parity error regardless of the Parity Error Response bit (CMD_STAT_REG.COMMAND_PARITY_ERROR_RESPONSE_ENABLE). |
| 30 | 0x0<br>RW/1C | **STATUS_SIGNALED_SYSTEM_ERROR:** Signaled System Error.<br>Never expected to be asserted. This bit is set when the device has detected an uncorrectable error and reported it via SERR message over sideband.<br>This requires SERR Enable bit to be set in Command register. |
| 29 | 0x0<br>RW/1C | **STATUS_RECEIVED_MASTER_ABORT:** Received Master Abort.<br>This bit is set when device receives a Completion transaction with Unsupported Request completion status.<br>No error will be reported. |
| 28 | 0x0<br>RW/1C | **STATUS_RECEIVED_TARGET_ABORT:** Received Target Abort.<br>This bit is set when device receives a Completion transaction with Completer Abort completion status.<br>No error will be reported. |
| 27 | 0x0<br>RW/1C | **STATUS_SIGNALED_TARGET_ABORT:** Signaled Target Abort.<br>Set by the device when aborting a request that violates the device programming model.<br>When SERR Enable is set SERR message will be send over sideband. |
| 26:25 | 0x0<br>RSV | **STATUS_RESERVED_25:** Reserved. |
| 24 | 0x0<br>RW/1C | **STATUS_MASTER_DATA_PARITY_ERROR:** Master Data Parity Error.<br>This bit is set if the Parity Error Response bit (CMD_STAT_REG.COMMAND_PARITY_ERROR_RESPONSE_ENABLE) is set and it receives a completion with poisoned data from IOSF. |
| 23:21 | 0x0<br>RSV | **STATUS_RESERVED_21:** Reserved. |
| 20 | 0x1<br>RO | **STATUS_NEW_CAP:** New Capabilities.<br>Indicates that a device implements extended capabilities.<br>The Controller sets this bit, and implements a capabilities list. |
| 19 | 0x0<br>RO/V | **STATUS_INTERRUPT_STATUS:** Interrupt Status.<br>The Interrupt Status field is a RO field that indicates that an interrupt message is pending internally to the device. |
| 18:16 | 0x0<br>RSV | **STATUS_RESERVED_16:** Reserved. |
| 15:11 | 0x0<br>RSV | **COMMAND_RESERVED_11:** Reserved. |
| 10 | 0x0<br>RW | **COMMAND_INTERRUPT_DISABLE:** Interrupt Disable.<br>Controls the ability of a device to generate a legacy interrupt message.<br>When set, the device cannot generate legacy interrupt messages. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 9 | 0x0<br>RSV | **COMMAND_RESERVED_9:** Reserved. |
| 8 | 0x0<br>RW | **COMMAND_SERR_ENABLE:** SERR# Enable |
| 7 | 0x0<br>RSV | **COMMAND_RESERVED_7:** Reserved. |
| 6 | 0x0<br>RW | **COMMAND_PARITY_ERROR_RESPONSE_ENABLE:** Parity Error Response.<br>This bit controls the setting of STATUS_MASTER_DATA_PARITY_ERROR.<br>0 = Parity errors can be ignored by the device.<br>1 = Device can report parity errors. |
| 5:3 | 0x0<br>RSV | **COMMAND_RESERVED_3:** Reserved. |
| 2 | 0x0<br>RW | **COMMAND_BUS_ACCESS_ENABLE:** Must be set to 1 for the device to initiate memory read and write requests as a master.<br>This bit does not affect a device's ability to generate MSI message, Error Message, or completion cycles as a target. |
| 1 | 0x0<br>RW | **COMMAND_MEM_ACCESS_ENABLE:** Memory Access Enable |
| 0 | 0x0<br>RW | **COMMAND_IO_ACCESS_ENABLE:** When set, IO Space Decoding is enabled and IO transactions targeting the device are accepted. |

## 53.2.3 REV_CLASS_CODE—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 8h

**Default:** 07000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x70002<br>RO | **CLASS_CODE:** The class code is a RO hard-coded value that identifies its functionality.<br>***Note:*** This field (Programming Interface, Sub Class Code, Base Class Code) is RO from Primary and RW from Sideband. |
| 7:0 | 0x0<br>RO/V | **REVISION_ID:** The default revision ID is 0x0. |

## 53.2.4 CACHE—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RSV | **BIST:** Reserved. |
| 23:16 | 0x80<br>RO | **HEADER_TYPE:** This indicates if a device is single-function or multifunction.<br>If a single function is the only active one then this field has a value of 0x00 to indicate a single-function device.<br>If other functions are enabled then this field has a value of 0x80 to indicate a multi-function device. |
| 15:8 | 0x0<br>RSV | **RESERVED_1:** Reserved. |
| 7:0 | 0x0<br>RSV | **RESERVED_0:** Reserved. |

## 53.2.5 IOBA—Offset 10h

The Base Address registers (BARs) are used to map register space of the various functions. It has a memory BAR, IO BAR.

The internal registers memories are accessed as direct memory mapped offsets from the Base Address register. Software can access a Dword or 64 bits.

*Note:* If this is a Dummy function the register is RO as 0.

**Type:** PCI Configuration Register (Size: 32 bits) **Offset:** [B:0, D:26, F:0] + 10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **BASE_ADDR:** Reserved. |
| 2:1 | 0x0<br>RO | **MEMORY_TYPE:** Hardwired to 0 for 8 byte access. |
| 0 | 0x1<br>RO | **MEMORY_IO_SPACE:** Reserved. |

## 53.2.6 MEMBA—Offset 14h

The Base Address registers (BARs) are used to map register space of the various functions.

The internal registers memories are accessed as direct memory mapped offsets from the Base Address register. Software can access a Dword or 64 bits.

**Type:** PCI Configuration Register (Size: 32 bits) **Offset:** [B:0, D:26, F:0] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **BASE_ADDR:** Reserved. |
| 7:4 | 0x0<br>RO | **MEM_SIZE:** Set to 0 to indicate 256 byte memory Space. |
| 3 | 0x0<br>RO | **PREFETCHABLE:** Set to 0 to indicate MEMBA is not prefetchable. |
| 2:1 | 0x0<br>RO | **MEMORY_TYPE:** Set to 0 to indicate the BAR is located in lower 4G space. |
| 0 | 0x0<br>RO | **MEMORY_IO_SPACE:** Reserved. |

## 53.2.7 SUBSYSTEM_VENDOR_ID—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 2Ch

**Default:** 00008086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O/P | **SUBSYSTEM_ID:**<br>***Note:*** This field is RO from Sideband and RW/O from Primary. |
| 15:0 | 0x8086<br>RW/O/P | **SUBSYSTEM_VENDOR_ID:**<br>***Note:*** This field is RO from Sideband and RW/O from Primary. |

## 53.2.8 CAP_PTR—Offset 34h

The Capabilities Pointer field is an 8 bit field that provides an offset in the device's PCI configuration space for the location of the first item in the Capabilities Linked List (CLL).

Controller sets this bit and implements a capabilities list to indicate that it supports PCI power management, Message Signaled Interrupts (MSIs), and PCIe extended capabilities. Its value is 0x40, which is the address of the first entry: MSI cap.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 34h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 7:0 | 0x40<br>RO | **CAP_PTR:** |

## 53.2.9 INTERRUPT—Offset 3Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 3Ch

**Default:** 000001FFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 15:8 | 0x1<br>RW/P/L | **INTERRUPT_PIN:** A value of 0x1, 0x2, 0x3, or 0x4 indicates that this function implements legacy interrupt on INTA#, INTB#, INTC#, or INTD#, respectively.<br>This bit is locked by PLKCTL.CL. |
| 7:0 | 0xff<br>RW | **INTERRUPT_LINE:** Read/write register programmed by software to indicate which of the system interrupt request lines this controller's interrupt pin is bound to.<br>See the PCIe definition for more details. Each of the PCIe functions has its own register.<br>***Note:*** For Dummy functions this register is RO as zero. |

## 53.2.10 MSIC—Offset 40h

MSI Capability Register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 40h

**Default:** 00005005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 23 | 0x0<br>RO | **AD64C:** 64-bit capable.<br>A value of 1 indicates it is capable of generating 64-bit message addresses. |
| 22:20 | 0x0<br>RW | **MMEN:** Multiple Message Enable.<br>Indicates it supports a single message per function. |
| 19:17 | 0x0<br>RO | **MMC:** Multiple Message Capable.<br>Indicates a single requested message per each function. |
| 16 | 0x0<br>RW | **MSI_EN:** If set to 1, equals MSI.<br>In this case, controller generates an MSI for interrupt assertion instead of INTx signaling. |
| 15:8 | 0x50<br>RO | **NEXT_POINTER:** Next cap Pointer.<br>This field provides an offset to the next capability item in the capability list.<br>Its value of 0x50 points to the PM capability structure. |
| 7:0 | 0x5<br>RO | **CAP_ID:** This field equals 0x05 indicating the linked list item as being the MSI registers. |

## 53.2.11 MSIA—Offset 44h

Written by the system to indicate 32 bits of the address to use for the MSI memory write transaction.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **MSIAddr:** Reserved. |
| 1:0 | 0x0<br>RO | **MSI_DW:** Value of 0 indicates memory address is always DW aligned. |

## 53.2.12 MSID—Offset 48h

Written by the system to indicate the lower 16 bits of the data written in the MSI memory write Dword transaction. The upper 16 bits of the transaction are written as 0b.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 15:0 | 0x0<br>RW | **MSID:** |

## 53.2.13 PMC—Offset 50h

All fields are reset on full power-up. All of the fields except PME_En and PME_Status are reset on exit from D3cold state. If aux power is not supplied, the PME_En and PME_Status fields also reset on exit from D3cold state.

*Note:* See the detailed description for registers loaded from the EEPROM at initialization time. Behavior of some fields depend on the Power Management bit in EEPROM word 0x0A.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 50h

**Default:** 00030001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **PME_SUPPORT:** This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. |
| 26 | 0x0<br>RO | **D2_SUPPORT:** Does not support D2 state. |
| 25 | 0x0<br>RO | **D1_SUPPORT:** Does not support D1 state. |
| 24:22 | 0x0<br>RO | **AUX_CURRENT:** Required current defined in the Data Register. |
| 21 | 0x0<br>RO | **DSI:** Controller requires its device driver to be executed following transition to the D0 uninitialized state. |
| 20 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 19 | 0x0<br>RO | **PME_CLOCK:** Disabled; Hardwired to 0. |
| 18:16 | 0x3<br>RO | **VERSION:** Complies with the PCI PM specification, revision 1.2 |
| 15:8 | 0x0<br>RO | **NEXT_POINTER:** This field provides an offset to the next capability item in the capability list. Value of 0xA0 points to the PCI Express capability. |
| 7:0 | 0x1<br>RO | **CAP_ID:** This fields equals 0x01 indicating the linked list item as being the PCI Power Management registers. |

## 53.2.14 PMCSR—Offset 54h

All fields are reset on full power-up. All of the fields except PME_En and PME_Status are reset on exit from D3cold state. If aux power is not supplied, the PME_En and PME_Status fields also reset on exit from D3cold state. Each device function has its own PMCSR register.

*Note:* See the detailed description for registers loaded from the EEPROM at initialization time. Behavior of some fields depend on the Power Management bit in EEPROM word 0x0A.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 54h

**Default:** 00000008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **DATA:** No Data. |
| 23:16 | 0x0<br>RSV | **RESERVED16:** Reserved. |
| 15 | 0x0<br>RO | **PME_STATUS:** |
| 14:9 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 8 | 0x0<br>RO | **PME_EN:** Hardwired to 0, indicate not support PME. |
| 7:4 | 0x0<br>RSV | **RESERVED4:** Reserved. |
| 3 | 0x1<br>RO | **NO_SOFT_RESET:**<br>• When set (1) this bit indicates that when controller transitions from D3hot to D0 because of modifying Power State bits in the PMCSR register, no internal reset is issued and Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the Power State bits.<br>• When clear (0) controller performs an internal reset upon transitioning from D3hot to D0 via software control of the Power State bits in the PMCSR register. Configuration Context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full re-initialization sequence is needed to return the device to D0 Initialized.<br>Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled. |
| 2 | 0x0<br>RSV | **RESERVED2:** Reserved. |
| 1:0 | 0x0<br>RW/V | **POWER_STATE:** This field is used to set and report the power state of a function as follows:<br>• 00b = D0<br>• 01b = D1 (cycle ignored if written with this value)<br>• 10b = D2 (cycle ignored if written with this value)<br>• 11b = D3 hot |

## 53.2.15 DSC—Offset 60h

Device Specific control and Status Register.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RSV | **RESERVED_3:** Reserved. |
| 19 | 0x0<br>RW/1C | **URD:** Unsupported Request Detect:<br>When set, this bit is set when unsupported request is detected. |
| 18:16 | 0x0<br>RSV | **RESERVED_2:** Reserved. |
| 15:4 | 0x0<br>RSV | **RESERVED_1:** Reserved. |
| 3 | 0x0<br>RW | **URRE:** Unsupported Request Reporting Enable:<br>When set, this bit enables the reporting of Non-fatal error. |
| 2:0 | 0x0<br>RSV | **RESERVED_0:** Reserved. |

## 53.2.16 FUNC_RDCFG_HIDE—Offset 74h

Function-Hide Register. Hiding is programmable per-function. Since hiding affects reads to FUNC_RDCFG_HIDE itself, this register is not shared in order to avoid read-modify-write errors.

*Note:* If function 0 is hidden, O/S will not discover function 1 and function 2.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:26, F:0] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RSV | **RESERVED_0:** Reserved. |
| 0 | 0x0<br>RW | **HIDE:** HIDE:<br>• 0 - HSUART Not Hidden<br>• 1 - HSUART Hidden |

## 53.3 Registers in Memory Space—MEMBA

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) MEMBA. There is a set of these registers for each of the three HSUART Functions (F0, F1, and F2).

### 53.3.1 RBR_THR_DLL—Offset 0h

RBR = Receive buffer register THR = Transmit hold register DLL = Divisor Latch low.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 0h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **RBR_THR_DLL:** When DLAB =0, this 8 bit is used as Received Buffer Register as Read, and is used as Transmit Holding Register as Write.<br>When DLAB is 1, this register is used as DLL (Divisor Latch Low Byte Register). |

### 53.3.2 IER_DLH—Offset 1h

IER = Interrupt Enable Register DLH = Divisor Latch High.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 1h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **IER_DLH:** When DLAB =0, this 8 bit is used as Interrupt Enable Register.<br>When DLAB =1, this 8 bit is used as DLH (Divisor Latch High Byte Register). |

## 53.3.3 IIR_FCR—Offset 2h

IIR = Interrupt Identification Register FCR = FIFO control register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 2h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **IIR_FCR:** When this 8 bit is read, it is serving as Interrupt Identification Registers.<br>When this 8 bit is written, it is serving as FCR (FIFO control register.). |

## 53.3.4 LCR—Offset 3h

Line Control Register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 3h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **LCR_DLAB:** Divisor Latch Register Access Bit:<br>The divisor-latch access bit must be set high (logic 1) to access the divisor latches of the baud rate generator during a read or write operation. It must be set low (logic 0) to access the receiver buffer, the Transmit Holding register, or the Interrupt Enable register.<br>This bit does not need to be set when using auto-baud.<br>• 0: Access Transmit Holding register (THR), Receive Buffer register (RBR) and Interrupt Enable register.<br>• 1: Access Divisor Latch registers (DLL and DLH). |
| 6 | 0x0<br>RW | **LCR_SB:** Set Break:<br>The set-break control bit transmits a Break condition to the receiving UART.<br>When SB is set to logic 1, the serial output (TXD) is forced to the Spacing (logic 0) state and remains there until SB is set to logic 0. This bit acts only on the TXD pin and has no effect on the transmitter logic.<br>• 0: No effect on TXD output.<br>• 1: Forces TXD output to 0 (space).<br>This feature enables the processor to alert a terminal in a computer communications system.<br>If the following sequence is executed, no erroneous characters will be transmitted because of the break:<br>Load 0x00 in the Transmit Holding register in response to a TDRQ Interrupt.<br>After TDRQ goes high (indicating that 0x00 is being shifted out), set the break bit before the parity or stop bits reach the TXD pin.<br>Wait for the transmitter to be idle (TEMT = 1), and clear the break bit when normal transmission has to be restored.<br>During the Break, the transmitter can be used as a character timer to accurately establish the Break duration.<br>In FIFO mode, wait for the transmitter to be idle (TEMT=1) to set and clear the break bit. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **LCR_STKYP:** Sticky Parity:<br>This bit is the sticky-parity bit, which can be used in multiprocessor communications.<br>• 0: No effect on parity bit.<br>• 1: Forces parity bit to be opposite of EPS bit value.<br>When PEN and STKYP are logic 1, the bit that is transmitted in the parity-bit location (the bit just before the stop bit) is the complement of the EPS bit. If EPS is 0, then the bit at the parity-bit location will be transmitted as a 1.<br>In the receiver, if STKYP and PEN are 1, then the receiver compares the bit that is received in the parity-bit location with the complement of the EPS bit.<br>If the values being compared are not equal, the receiver sets the parity-error bit in LSR, and causes an Error Interrupt if Line-Status Interrupts were enabled.<br>For example, if EPS is 0, the receiver expects the bit received at the parity-bit location to be 1.<br>If it is not, then the parity-error bit is set.<br>By forcing the bit value at the parity-bit location, rather than calculating a parity value, a system with a master transmitter and multiple receivers can identify some transmitted characters as receiver addresses and the rest of the characters as data.<br>If PEN = 0, STKYP is ignored. |
| 4 | 0x0<br>RW | **LCR_EPS:** Even Parity Select:<br>EPS is only valid when the parity is enabled (PEN = 1).<br>• 0: Sends and checks for odd parity.<br>• 1: Sends and checks for even parity. |
| 3 | 0x0<br>RW | **LCR_PEN:** Parity Enable (PEN): When PEN is logic 1, a parity bit is generated (transmit data) or checked (receive data) between the last data-word bit and stop bit of the serial data.<br>• 0: No parity function.<br>• 1: Allows parity generation and checking. |
| 2 | 0x0<br>RW | **LCR_STB:** Stop Bits:<br>This bit specifies the number of stop bits transmitted and received in each serial character.<br>• 0: 1 stop bit.<br>• 1: 2 stop bits; except for 5-bit character, then 1 bits The receiver checks the first stop bit only, regardless of the number of stop bits selected. |
| 1:0 | 0x0<br>RW | **LCR_WLS:** Word Length Select:<br>The word length select bits specify the number of data bits (five to eight bits are allowed) in each transmitted or received serial character.<br>• 00: 5-bit character (default).<br>• 01: 6-bit character 10: 7-bit character 11: 8-bit character. |

## 53.3.5 MCR—Offset 4h

Modem Control Register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [MEMBA] + 4h

**Default:** 00h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RSV | **MCR_RESERVED:** Reserved. |
| 5 | 0x0<br>RW | **MCR_AFE:** MCR-Auto Flow Enable:<br>This is a reserved bit in 16550 spec.<br>• 0: Auto-RTS and Auto-CTS are disabled.<br>• 1: Auto-CTS is enabled. If MCR[RTS] is also set, both Auto-CTS and Auto-RTS is enabled.<br>Auto-RTS is not enabled if AFE is not set regardless of the state of MCR[RTS].<br>Autoflow automates the flow of data between the UART and the remote device. |
| 4 | 0x0<br>RW | **MCR_LOOP:** Modem Control Register-Loop back test mode.<br>This bit provides a local loopback feature for diagnostic testing of the UART.<br>This feature allows the processor to verify the UART transmit and receive data paths.<br>The Transmit, Receive, and Modem Control interrupts are operational in this mode.<br>• 0: Normal UART operation.<br>• 1: Loopback Test mode operation Serial output uart_txd pin is set to logic 1.<br>Serial input uart_rxd pin is disconnected from external input line.<br>The output of the Transmitter Shift register is looped back into the Receiver Shift register.<br>The modem-control inputs (uart_cts, uart_dsr, uart_dcd, and uart_ri) are activated by MCR bits 0-3 instead of the modem-control inputs.<br><br>A Break signal can also be transferred from the transmitter section to the receiver section in Loopback mode. When LOOP is set to logic 1, the following will occur:<br>The TXD (transmitter output) pin is set to a logic-1 state.<br>The RXD (receiver input) pin is disconnected.<br>The output of the Transmitter Shift register is looped back into the Receiver-Shift register input.<br>The four modem-control inputs (uart_cts, uart_dsr, uart_dcd, and uart_ri) are disconnected from the pins and the modem-control output pins (uart_rts and uart_dtr) are forced to their inactive state.<br>The lower four bits of the Modem Control register (MCR) are connected to the upper four modem status register (MSR) bits.<br>Flow control can be tested; if autoflow is enabled, the MCR[RTS] bit has no effect on the nCTS input as nRTS is asserted by the autoflow logic.<br>• DTR = 1 forces DSR to a 1<br>• RTS = 1 forces CTS to a 1<br>• OUT1 = 1 forces RI to a 1<br>• OUT2= 1 forces DCD to a 1<br>Coming out of the Loopback Test mode may result in unpredictable activation of the delta bits (bits 3:0) in the Modem Status register (MSR).<br>It is recommended that MSR is read once to clear the delta bits in the MSR. |
| 3 | 0x0<br>RW | **MCR_OUT2:** Output2:<br>The function of the auxiliary output OUT2 bit differs depending on the mode of the UART.<br>• 0: uart_dcd UART pin input set low when in Loop-back mode.<br>• 1: uart_dcd UART pin input set high when in Loop-back mode. |
| 2 | 0x0<br>RW | **MCR_OUT1:** Modem Control Register-output1:<br>The auxiliary output OUT1 bit is only used in Loopback Test mode.<br>• 0: uart_ri UART pin input set low when in Loop-back mode.<br>• 1: uart_ri UART pin input set high when in Loop-back mode. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **MCR_RTS:** When Auto Flow is disabled: MCR[AFE] = 0:<br>• 0: Primary output uart_rts pin is forced to logic 1.<br>• 1: Primary output uart_rts pin is forced to logic 0.<br>When Auto Flow is enabled: MCR[AFE] = 1:<br>• 0: Auto-RTS is disabled.<br>• 1: Auto-RTS is enabled. |
| 0 | 0x0<br>RW | **MCR_DTR:** Modem Control Register-Data Terminal Ready.<br>This bit controls the Data Terminal Ready output.<br>• 0: Primary output uart_dtr pin is forced to logic 1.<br>• 1: Primary output uart_dtr pin is forced to logic 0.<br>The uart_dtr output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding modem or data set. |

## 53.3.6 LSR—Offset 5h

Line Status Register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 60h

**BAR and Offset:** [MEMBA] + 5h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **LSR_FIFOE:** Line Status Register - FIFO Error Status:<br>• 0: No character has error inside the Receive FIFO.<br>• 1: At least one character in Receiver FIFO has errors.<br>In non-FIFO mode, this bit is always set to 0. In FIFO mode, FIFOE is set to 1 when there is at least one parity error, framing error, or break indication for any of the characters in the FIFO.<br>A processor read to the Line Status register does not reset this bit.<br>FIFOE is reset when all error bytes have been read from the FIFO.<br>If DMA requests are enabled and FIFOE is set to 1, the Error Interrupt is generated and no Receive DMA request is generated even if the Receive FIFO reaches the trigger level.<br>If DMA requests are not enabled setting FIFOE to 1 does not generate an interrupt. |
| 6 | 0x1<br>RO/V | **LSR_TEMT:** Line Status Register - Transmitter Empty:<br>• 0: There is data in the Transmit Shift register, the Transmit Holding register, or the FIFO (in FIFO mode).<br>• 1: All the data in the transmitter has been shifted out. |
| 5 | 0x1<br>RO/V | **LSR_TDRQ:** Line Status Register - Transmit data Request:<br>• 0: The UART is NOT ready to receive data for transmission.<br>• 1: The UART is ready to receive data for transmission.<br>The assertion of TDRQ causes the UART to interrupt the processor when the Transmit Data Request Interrupt Enable IER[TIE] is set high, and generates a DMA request to the DMA controller if DMA Request is enabled.<br>TDRQ is set to 1 when the new symbol is copied from Transmit Holding register into the Transmit Shift register.<br>The bit is reset to logic 0 concurrently with the loading of the Transmit Holding register by the processor.<br>In FIFO mode, TDRQ is set to logic 1 when the FIFO is less than half full.<br>It is cleared when the FIFO is more than half full.<br>If more than 64 characters are loaded into the FIFO, the excess characters are lost. |
| 4 | 0x0<br>RO/C/V | **LSR_BI:** Line Status Register - Break Interrupt:<br>• 0: No break signal has been received.<br>• 1: Break signal has been received.<br>BI is set to logic 1 when the received data input is held in the Spacing (logic 0) state for longer than a full character transmission time (that is, the total time of start bit + data bits + parity bit + stop bits).<br>The Break Indicator is reset when the processor reads the Line Status register. In FIFO mode, only one Break character (equal to 0x00) is loaded into the FIFO regardless of the length of the Break condition.<br>BI shows the Break condition for the character at the bottom of the FIFO, not the most recent character received. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RO/C/V | **LSR_FE:** Line Status Register - frame error:<br>• 0: No Framing error.<br>• 1: Framing error has occurred.<br>FE indicates that the received character did not have a valid stop bit. FE is set to logic 1 when the bit following the last data bit or parity bit is detected as logic 0 bit (spacing level).<br>If the Line Control register had been set for two stop bits, the receiver does not check for a valid second stop bit.<br>The FE indicator is reset when the processor reads the Line Status register.<br>The UART will resynchronize after a framing error by assuming that the framing error was due to the next start bit.<br>Therefore it samples this start bit twice and then takes in the data. In FIFO mode, FE shows a framing error for the character at the bottom of the FIFO, not for the most recent character received. |
| 2 | 0x0<br>RO/C/V | **LSR_PE:** Line Status Register - Parity Enable:<br>• 0: No Parity error.<br>• 1: Parity error has occurred.<br>PE indicates that the received character does not have the correct even or odd parity, as selected by the Even Parity Select bit.<br>The PE is set to logic 1 upon detection of a parity error and is reset to logic 0 when the processor reads the Line Status register.<br>In FIFO mode, PE shows a parity error for the character at the bottom of the FIFO, not the most recent character received. |
| 1 | 0x0<br>RO/C/V | **LSR_OE:** Line Status Register - Overflow Error:<br>• 0: No overflow error. Data has not been lost.<br>• 1: Overflow error. Receive data has been lost.<br>In non-FIFO mode, OE indicates that data in the Receiver Buffer register was not read by the processor before the next character was received; the new character is lost.<br>In FIFO mode, OE indicates that all 64 bytes of the FIFO are full and the most recently received byte has been discarded.<br>The OE indicator is set to logic 1 upon detection of an overflow condition and reset when the processor reads the Line Status register. |
| 0 | 0x0<br>RO/V | **LSR_DR:** Line Status Register - Data Ready:<br>• 0: No data has been received.<br>• 1: Data is available in RBR or the FIFO.<br>DR is set to logic 1 when a complete incoming character has been received and transferred into the Receiver Buffer register or the FIFO. In non-FIFO mode, DR is reset to 0 when the receive buffer is read.<br>In FIFO mode, DR is reset to logic 0 if the FIFO is empty (last character has been read from RBR) or the RESETRF bit is set in FCR. |

## 53.3.7 MSR—Offset 6h

Modem Status Register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 6h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **MSR_DCD:** Modem Status Register - data Carrier Detect:<br>Loopback is disabled, MCR[LOOP] = 0.<br>• 0: Primary input uart_dcd is logic 1.<br>• 1: Primary input uart_dcd is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1.<br>• 0: MCR[OUT2] is logic 0.<br>• 1: MCR[OUT2] is logic 1. |
| 6 | 0x0<br>RO/V | **MSR_RI:** Modem Status Register - Ring Indicator:<br>Loopback is disabled, MCR[LOOP] = 0.<br>• 0: Primary input uart_ri is logic 1.<br>• 1: Primary input uart_ri is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1.<br>• 0: MCR[OUT1] is logic 0.<br>• 1: MCR[OUT1] is logic 1. |
| 5 | 0x0<br>RO/V | **MSR_DSR:** Modem Status Register - Data Set Ready:<br>Loopback is disabled, MCR[LOOP] = 0.<br>• 0: Primary input uart_dsr is logic 1.<br>• 1: Primary input uart_dsr is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1.<br>• 0: MCR[DTR] is logic 0.<br>• 1: MCR[DTR] is logic 1. |
| 4 | 0x0<br>RO/V | **MSR_CTS:** Modem Status Register - Clear to Send:<br>Loopback is disabled, MCR[LOOP] = 0.<br>• 0: Primary input uart_cts is logic 1.<br>• 1: Primary input uart_cts is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1.<br>• 0: MCR[RTS] is logic 0.<br>• 1: MCR[RTS] is logic 1. |
| 3 | 0x0<br>RO/C/V | **MSR_DDCR:** Modem Status Register - Delta Data Carrier Ready:<br>• 0: No change in uart_dcr pin since last read of MSR.<br>• 1: uart_dcr pin has changed state.<br>When DDCR is set the modem status interrupt will be generated if enabled in IER. |
| 2 | 0x0<br>RO/C/V | **MSR_TERI:** Modem Status Register - Trailing Edge Ring Indicator:<br>• 0: uart_ri pin has not changed from 0 to 1 since last read of MSR.<br>• 1: uart_ri pin has changed from 0 to 1.<br>When TERI is set the modem status interrupt will be generated if enabled in IER. |
| 1 | 0x0<br>RO/C/V | **MSR_DDSR:** Modem Status Register - Delta Data Set Ready:<br>• 0: No change in uart_dsr pin since last read of MSR.<br>• 1: uart_dsr pin has changed state.<br>When DDSR is set the modem status interrupt will be generated if enabled in IER. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **MSR_DCTS:** Modem Status Register - Delta Clear to Send:<br>• 0: No change in uart_cts pin since last read of MSR.<br>• 1: uart_cts pin has change state.<br>When DCTS is set the modem status interrupt will be generated if enabled in IER. |

## 53.3.8 SPR—Offset 7h

Scratch Pad Status Register.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [MEMBA] + 7h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **SPR:** Scratch Pad Register (SCRATCH):<br>Writing to this field does not affect the operation of the UART in any way. |

## 53.3.9 UART_FISR—Offset 8h

UART Functional Interrupt Status Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + 8h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 2 | 0x0<br>RO/V | **DWCIS:** DMA Write Channel Interrupt Status (DWCIS):<br>Set when Functions DMA write channel interrupt line is asserted. |
| 1 | 0x0<br>RO/V | **DRCIS:** DMA Read Channel Interrupt Status (DRCIS):<br>Set when Functions DMA read channel interrupt line is asserted. |
| 0 | 0x0<br>RO/V | **UIS:** UART Interrupt Status (UIS):<br>Set when Functions UART interrupt line is asserted. |

## 53.3.10 RFOR—Offset 20h

Receiver FIFO Occupancy Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + 20h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 6:0 | 0x0<br>RO | **RFBC:** Receive FIFO Byte Count (RFBC):<br>Number of bytes (0-64) present in the receive FIFO. |

## 53.3.11 ABR—Offset 24h

Auto BAUD Control Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000010h

**BAR and Offset:** [MEMBA] + 24h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 4 | 0x1<br>RW | **UUE:** UART Unit Enable (UUE):<br>• 0: UART transceiver is disabled.<br>• 1: UART transceiver is enabled.<br>Transceiver is also disabled by hardware whenever the function is D3-hot PM state. |
| 3 | 0x0<br>RW | **ABT:** Auto-baud Table (ABT):<br>• 0: Formula used to calculate baud rates allowing all possible baud rates.<br>• 1: Standard baud rates to be chosen by UART.<br>When set to 1, UART will round the calculated baud rate to nearest standard baud rate. |
| 2 | 0x0<br>RW | **ABUP:** Auto-Baud UART Program (ABUP):<br>• 0: Processor Programs Divisor Latch registers.<br>• 1: UART Programs Divisor Latch registers Setting ABUP allows the UART to automatically program the Divisor Latch registers (DLL and DLH) when the auto-baud circuit has detected the baud rate, regardless of the state of the Divisor-Latch Access bit (LCR[DLAB]). For this to occur, the Auto-Baud Enable (ABE) bit must be set.<br>Clearing this bit allows the processor to read the Auto-Baud Count register (ACR) and determine the baud rate using its own algorithm rather than using the UARTs. |
| 1 | 0x0<br>RW | **ABLIE:** Auto-Baud Lock Interrupt Enable (ABLIE):<br>• 0: Auto Baud Lock Interrupt disabled (Source IIR[ABL]).<br>• 1: Auto Baud Lock Interrupt enabled (Source IIR[ABL]).<br>Setting ABLIE enables the IIR[ABL] Interrupt, which occurs when the auto-baud circuit has detected the baud rate and written the value into the Auto-Baud Count register (ACR).<br>The Divisor Latch registers can then be programmed by the processor or auto-baud circuitry as dictated by the state of the ABUP. |
| 0 | 0x0<br>RW | **ABE:** Auto-Baud Enable (ABE):<br>• 0: Auto Baud disabled.<br>• 1: Auto Baud enabled.<br>Setting ABE enables the auto-baud circuitry within the UART. The circuitry counts the number of clocks in the start bit and writes this count into the Auto-Baud Count register (ACR).<br>It will then interrupt the processor if the Auto Baud Lock Interrupt Enable (ABLIE) bit is set.<br>It will also automatically program the Divisor Latch registers (DLL and DLH) if the Auto-Baud UART Program (ABUP) bit is set. |

## 53.3.12 ACR—Offset 28h

Auto Baud Count Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + 28h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 15:0 | 0x0<br>RO/V | **ABC:** Auto-Baud Count (ABC):<br>Auto-baud count is the number of FUART clock cycles within a start bit pulse. |

## 53.3.13 PSR—Offset 30h

Pre-Scaler Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + 30h

**Default:** 00000010h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 15:0 | 0x10<br>RW | **PS:** Pre-Scalar (PS):<br># of samples per character bit. |

## 53.3.14 UCMR—Offset 34h

UART CLOCK GEN Multiplier Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + 34h

**Default:** 000000D8h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 23:0 | 0xd8<br>RW | **UCMUL:** UART Clock Gen Multiplier (UCMUL):<br>UART Clock Gen Multiplier to derive the UART transceiver clock. |

## 53.3.15 UCDR—Offset 38h

UART CLOCK GEN Divisor Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + 38h

**Default:** 00003D09h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RSV | **RESERVED:** Reserved. |
| 23:0 | 0x3d09<br>RW | **UCDIV:** UART Clock Gen Divisor (UCDIV):<br>UART Clock Gen Divisor to derive the UART transceiver clock. |

## 53.3.16 CH0SR—Offset 80h

Channel Status Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000F0000h

**BAR and Offset:** [MEMBA] + 80h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO/V | **CDESC:** Current Descriptor (CDESC):<br>• 00: Descriptor 0<br>• 01: Descriptor 1<br>• 10: Descriptor 2<br>• 11: Descriptor 3<br>The current descriptor will points to the descriptor to be served at any moment.<br>While none of the descriptor is valid the current descriptor will point to the descriptor 0. |
| 29:20 | 0x0<br>RSV | **RESERVED_2:** Reserved 2. |
| 19:16 | 0xf<br>RO/V | **DESCE:** Descriptor Empty (DESCE):<br>One bit per descriptor, value of 1 indicates the descriptor transfer size is not 0. |
| 15 | 0x0<br>RO/C/V | **CHE:** Channel Error (CHE):<br>When set this bit indicates the following error condition. write channel:<br>RX FIFO error status transitioned from 1 to 0.<br>Once read this bit will only get set again when the RX FIFO Error status transition from 1 to 0.<br>Error status is assumed 1 when FIFO contains erroneous data. read channel:<br>The last read request is returned with malformed completion. In this case the UART TX FIFO will not be updated. |
| 14:12 | 0x0<br>RSV | **RESERVED_1:** Reserved 1. |
| 11:8 | 0x0<br>RO/C/V | **DESCTO:** Descriptor Time Out (DESCTO):<br>One bit per descriptor, value of 1 indicates the descriptor has timed out.<br>This only valid for write channel. In the event of descriptor timeout the available data in the buffer will be written to the memory and channel timeout interrupt will be asserted if enabled through descriptor control register.<br>If the timeout condition will occur at the descriptor done boundary only descriptor done status is updated. |
| 7:4 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 3:0 | 0x0<br>RO/C/V | **DESCD:** Descriptor Done (DESCD):<br>One bit per descriptor, value of 1 indicates the descriptor request has been completed. |

## 53.3.17 CH0CR—Offset 84h

Channel Control Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + 84h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 1 | 0x0<br>WO | **CHD:** Channel Direction (CHD):<br>• 0: Read Channel (UART TX).<br>• 1: Write Channel (UART RX). |
| 0 | 0x0<br>WO | **CHA:** Channel Active (CHA):<br>• 0: Channel is disabled.<br>• 1: Channel is enabled.<br>Whenever channel stops the hardware will clear this bit. Channel can be stopped by HW on descriptor done, timeout or error condition. |

## 53.3.18 CH0DCR—Offset 88h

Channel 0 Descriptor Control Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + 88h

**Default:** 00008000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RSV | **RESERVED_3:** Reserved 3. |
| 27:24 | 0x0<br>RW | **CHTOI:** Channel Time Out Interrupt (CHTOI):<br>One bit per descriptor, value of 1 indicates the channel should generate interrupt once timeout is detected. This is only applicable to write channel. |
| 23 | 0x0<br>RW | **CHEI:** Channel Error Interrupt (CHEI):<br>Value of 1 indicates the channel should generate interrupt if the channel error condition has occurred. |
| 22:20 | 0x0<br>RSV | **RESERVED_2:** Reserved 2. |
| 19:16 | 0x0<br>RW | **CHDI:** Channel Done Interrupt (CHDI):<br>One bit per descriptor, value of 1 indicates the channel should generate interrupt once descriptor is done. |
| 15 | 0x1<br>RW | **CHSOE:** Channel Stop on Error (CHSOTO):<br>Value of 1 indicates the channel should stop on error condition while any serving descriptor. |
| 14 | 0x0<br>RW | **CHSOTO:** Channel Stop on Timeout (CHSOTO):<br>Value of 1 indicates the channel should stop on timeout condition while any serving descriptor. |
| 13:12 | 0x0<br>RSV | **RESERVED_1:** Reserved 1. |
| 11:8 | 0x0<br>RW | **CHSOD:** Channel Stop on Done (CHSOD):<br>One bit per descriptor, value of 1 indicates the channel should stop once the descriptor request has been completed. |
| 7:4 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 3:0 | 0x0<br>RW/V | **DESCA:** Descriptor Active (DESCA):<br>One bit per descriptor, value of 1 indicates the descriptor is active.<br>Whenever the descriptor is done or timed out the corresponding descriptor active bit will be cleared. |

## 53.3.19 CH0BSR—Offset 90h

Channel Buffer Size.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + 90h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW | **CHBS:** Channel Buffer Size (CHBS):<br>Indicates the size of UART FIFO connected to the channel. |

## 53.3.20 CH0MTSR—Offset 94h

Channel Minimum Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + 94h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW | **CHMTSR:** Channel Minimum Transfer Size (CHMTSR):<br>Indicates the minimum number of bytes required to make the DMA request. |

## 53.3.21 CH0D0SAR—Offset A0h

Channel Descriptor 0 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + A0h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD0SAR:** Channel Descriptor 0 Start Address (CHD0SAR): |

## 53.3.22 CH0D0TSR—Offset A4h

Channel Descriptor 0 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + A4h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD0TS:** Channel Descriptor 0 Transfer Size (CHD0TS): |

## 53.3.23 CH0D1SAR—Offset A8h

Channel Descriptor 1 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + A8h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD1SAR:** Channel Descriptor 1 Start Address (CHD1SAR): |

## 53.3.24 CH0D1TSR—Offset ACh

Channel Descriptor 1 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + ACh

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD1TS:** Channel Descriptor 1 Transfer Size (CHD1TS): |

## 53.3.25 CH0D2SAR—Offset B0h

Channel Descriptor 2 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + B0h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD2SAR:** Channel Descriptor 2 Start Address (CHD2SAR): |

## 53.3.26 CH0D2TSR—Offset B4h

Channel Descriptor 2 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + B4h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD2TS:** Channel Descriptor 2 Transfer Size (CHD2TS): |

## 53.3.27 CH0D3SAR—Offset B8h

Channel Descriptor 3 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + B8h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD3SAR:** Channel Descriptor 3 Start Address (CHD3SAR): |

## 53.3.28 CH0D3TSR—Offset BCh

Channel Descriptor 3 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + BCh

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD3TS:** Channel Descriptor 3 Transfer Size (CHD3TS): |

## 53.3.29 CH1SR—Offset C0h

Channel Status Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000F0000h

**BAR and Offset:** [MEMBA] + C0h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO/V | **CDESC:** Current Descriptor (CDESC):<br>• 00: Descriptor 0<br>• 01: Descriptor 1<br>• 10: Descriptor 2<br>• 11: Descriptor 3<br>The current descriptor will points to the descriptor to be served at any moment.<br>While none of the descriptor is valid the current descriptor will point to the descriptor 0. |
| 29:20 | 0x0<br>RSV | **RESERVED_2:** Reserved 2. |
| 19:16 | 0xf<br>RO/V | **DESCE:** Descriptor Empty (DESCE):<br>One bit per descriptor, value of 1 indicates the descriptor transfer size is not 0. |
| 15 | 0x0<br>RO/C/V | **CHE:** Channel Error (CHE):<br>When set this bit indicates the following error condition. write channel: RX FIFO error status transitioned from 1 to 0.<br>Once read this bit will only get set again when the RX FIFO Error status transition from 1 to 0.<br>Error status is assumed 1 when FIFO contains erroneous data. read channel: The last read request is returned with malformed completion.<br>In this case the UART TX FIFO will not be updated. |
| 14:12 | 0x0<br>RSV | **RESERVED_1:** Reserved 1. |
| 11:8 | 0x0<br>RO/C/V | **DESCTO:** Descriptor Time Out (DESCTO):<br>One bit per descriptor, value of 1 indicates the descriptor has timed out.<br>This only valid for write channel.<br>In the event of descriptor timeout the available data in the buffer will be written to the memory and channel timeout interrupt will be asserted if enabled through descriptor control register.<br>If the timeout condition will occur at the descriptor done boundary only descriptor done status is updated. |
| 7:4 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 3:0 | 0x0<br>RO/C/V | **DESCD:** Descriptor Done (DESCD):<br>One bit per descriptor, value of 1 indicates the descriptor request has been completed. |

## 53.3.30 CH1CR—Offset C4h

Channel Control Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + C4h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 1 | 0x0<br>WO | **CHD:** Channel Direction (CHD):<br>• 0: Read Channel (UART TX).<br>• 1: Write Channel (UART RX). |
| 0 | 0x0<br>WO | **CHA:** Channel Active (CHA):<br>• 0: Channel is disabled.<br>• 1: Channel is enabled.<br>Whenever channel stops the hardware will clear this bit.<br>Channel can be stopped by HW on descriptor done, timeout or error condition. |

## 53.3.31 CH1DCR—Offset C8h

Channel 0 Descriptor Control Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00008000h

**BAR and Offset:** [MEMBA] + C8h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RSV | **RESERVED_3:** Reserved 3. |
| 27:24 | 0x0<br>RW | **CHTOI:** Channel Time Out Interrupt (CHTOI):<br>One bit per descriptor, value of 1 indicates the channel should generate interrupt once timeout is detected.<br>This is only applicable to write channel. |
| 23 | 0x0<br>RW | **CHEI:** Channel Error Interrupt (CHEI):<br>Value of 1 indicates the channel should generate interrupt if the channel error condition has occurred. |
| 22:20 | 0x0<br>RSV | **RESERVED_2:** Reserved 2. |
| 19:16 | 0x0<br>RW | **CHDI:** Channel Done Interrupt (CHDI):<br>One bit per descriptor, value of 1 indicates the channel should generate interrupt once descriptor is done. |
| 15 | 0x1<br>RW | **CHSOE:** Channel Stop on Error (CHSOTO):<br>Value of 1 indicates the channel should stop on error condition while any serving descriptor. |
| 14 | 0x0<br>RW | **CHSOTO:** Channel Stop on Timeout (CHSOTO):<br>Value of 1 indicates the channel should stop on timeout condition while any serving descriptor. |
| 13:12 | 0x0<br>RSV | **RESERVED_1:** Reserved 1. |
| 11:8 | 0x0<br>RW | **CHSOD:** Channel Stop on Done (CHSOD):<br>One bit per descriptor, value of 1 indicates the channel should stop once the descriptor request has been completed. |
| 7:4 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 3:0 | 0x0<br>RW/V | **DESCA:** Descriptor Active (DESCA):<br>One bit per descriptor, value of 1 indicates the descriptor is active.<br>Whenever the descriptor is done or timed out the corresponding descriptor active bit will be cleared. |

## 53.3.32 CH1BSR—Offset D0h

Channel Buffer Size.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + D0h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW | **CHBS:** Channel Buffer Size (CHBS):<br>Indicates the size of UART FIFO connected to the channel. |

## 53.3.33 CH1MTSR—Offset D4h

Channel Minimum Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + D4h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW | **CHMTSR:** Channel Minimum Transfer Size (CHMTSR):<br>Indicates the minimum number of bytes required to make the DMA request. |

## 53.3.34 CH1D0SAR—Offset E0h

Channel Descriptor 0 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + E0h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD0SAR:** Channel Descriptor 0 Start Address (CHD0SAR): |

## 53.3.35 CH1D0TSR—Offset E4h

Channel Descriptor 0 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + E4h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD0TS:** Channel Descriptor 0 Transfer Size (CHD0TS): |

## 53.3.36 CH1D1SAR—Offset E8h

Channel Descriptor 1 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + E8h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD1SAR:** Channel Descriptor 1 Start Address (CHD1SAR): |

## 53.3.37 CH1D1TSR—Offset ECh

Channel Descriptor 1 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + ECh

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD1TS:** Channel Descriptor 1 Transfer Size (CHD1TS): |

## 53.3.38 CH1D2SAR—Offset F0h

Channel Descriptor 2 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + F0h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD2SAR:** Channel Descriptor 2 Start Address (CHD2SAR): |

## 53.3.39 CH1D2TSR—Offset F4h

Channel Descriptor 2 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [MEMBA] + F4h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD2TS:** Channel Descriptor 2 Transfer Size (CHD2TS): |

## 53.3.40 CH1D3SAR—Offset F8h

Channel Descriptor 3 Start Address Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + F8h

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **CHD3SAR:** Channel Descriptor 3 Start Address (CHD3SAR): |

## 53.3.41 CH1D3TSR—Offset FCh

Channel Descriptor 3 Transfer Size Register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [MEMBA] + FCh

**Default:** 00000000h

**MEMBA Type:** PCI Configuration Register (Size: 32 bits)
**MEMBA Reference:** [B:0, D:26, F:0,1,2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RSV | **RESERVED_0:** Reserved 0. |
| 15:0 | 0x0<br>RW/V | **CHD3TS:** Channel Descriptor 3 Transfer Size (CHD3TS): |

## 53.4 Registers in I/O Space—IOBA

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IOBA described in Section 53.2.5, "IOBA—Offset 10h" on page 2391. There is a set of these registers for each of the three HSUART Functions (F0, F1, and F2).

### 53.4.1 RBR_THR_DLL—Offset 0h

RBR = Receive buffer register THR = Transmit hold register DLL = Divisor Latch low.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IOBA] + 0h

**Default:** 00h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **RBR_THR_DLL:**<br>When DLAB =0, this 8 bit is used as Received Buffer Register as Read, and is used as Transmit Holding Register as Write.<br>When DLAB is 1, this register is used as DLL (Divisor Latch Low Byte Register). |

### 53.4.2 IER_DLH—Offset 1h

IER = Interrupt Enable Register DLH = Divisor Latch High.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IOBA] + 1h

**Default:** 00h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **IER_DLH:**<br>When DLAB =0, this 8 bit is used as Interrupt Enable Register.<br>When DLAB =1, this 8 bit is used as DLH (Divisor Latch High Byte Register). |

## 53.4.3 IIR_FCR—Offset 2h

IIR = Interrupt Identification Register FCR = FIFO control register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 2h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **IIR_FCR:**<br>When this 8 bit is read, it is serving as Interrupt Identification Registers.<br>When this 8 bit is written, it is serving as FCR (FIFO control register.). |

## 53.4.4 LCR—Offset 3h

Line Control Register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 3h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **LCR_DLAB:** Divisor Latch Register Access Bit (DLAB):<br>The divisor-latch access bit must be set high (logic 1) to access the divisor latches of the baud rate generator during a read or write operation.<br>It must be set low (logic 0) to access the receiver buffer, the Transmit Holding register, or the Interrupt Enable register.<br>This bit does not need to be set when using auto-baud.<br>• 0: Access Transmit Holding register (THR), Receive Buffer register (RBR) and Interrupt Enable register.<br>• 1: Access Divisor Latch registers (DLL and DLH) |
| 6 | 0x0<br>RW | **LCR_SB:** Set Break (SB):<br>The set-break control bit transmits a Break condition to the receiving UART. When SB is set to logic 1, the serial output (TXD) is forced to the Spacing (logic 0) state and remains there until SB is set to logic 0.<br>This bit acts only on the TXD pin and has no effect on the transmitter logic.<br>• 0: No effect on TXD output.<br>• 1: Forces TXD output to 0 (space).<br>This feature enables the processor to alert a terminal in a computer communications system.<br>If the following sequence is executed, no erroneous characters will be transmitted because of the break:<br>Load 0x00 in the Transmit Holding register in response to a TDRQ Interrupt.<br>After TDRQ goes high (indicating that 0x00 is being shifted out), set the break bit before the parity or stop bits reach the TXD pin.<br>Wait for the transmitter to be idle (TEMT = 1), and clear the break bit when normal transmission has to be restored.<br>During the Break, the transmitter can be used as a character timer to accurately establish the Break duration.<br>In FIFO mode, wait for the transmitter to be idle (TEMT=1) to set and clear the break bit. |
| 5 | 0x0<br>RW | **LCR_STKYP:** Sticky Parity (STKYP):<br>This bit is the sticky-parity bit, which can be used in multiprocessor communications.<br>• 0: No effect on parity bit.<br>• 1: Forces parity bit to be opposite of EPS bit value.<br>When PEN and STKYP are logic 1, the bit that is transmitted in the parity-bit location (the bit just before the stop bit) is the complement of the EPS bit.<br>If EPS is 0, then the bit at the parity-bit location will be transmitted as a 1.<br>In the receiver, if STKYP and PEN are 1, then the receiver compares the bit that is received in the parity-bit location with the complement of the EPS bit.<br>If the values being compared are not equal, the receiver sets the parity-error bit in LSR, and causes an Error Interrupt if Line-Status Interrupts were enabled.<br>For example, if EPS is 0, the receiver expects the bit received at the parity-bit location to be 1.<br>If it is not, then the parity-error bit is set. By forcing the bit value at the parity-bit location, rather than calculating a parity value, a system with a master transmitter and multiple receivers can identify some transmitted characters as receiver addresses and the rest of the characters as data.<br>If PEN = 0, STKYP is ignored. |
| 4 | 0x0<br>RW | **LCR_EPS:** Even Parity Select (EPS):<br>EPS is only valid when the parity is enabled (PEN = 1).<br>• 0: Sends and checks for odd parity.<br>• 1: Sends and checks for even parity. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW | **LCR_PEN:** Parity Enable (PEN):<br>When PEN is logic 1, a parity bit is generated (transmit data) or checked (receive data) between the last data-word bit and stop bit of the serial data.<br>• 0: No parity function.<br>• 1: Allows parity generation and checking. |
| 2 | 0x0<br>RW | **LCR_STB:** Stop Bits (STB):<br>This bit specifies the number of stop bits transmitted and received in each serial character.<br>• 0: 1 stop bit<br>• 1: 2 stop bits; except for 5-bit character, then 1 bits The receiver checks the first stop bit only, regardless of the number of stop bits selected. |
| 1:0 | 0x0<br>RW | **LCR_WLS:** Word Length Select (WLS):<br>The word length select bits specify the number of data bits (five to eight bits are allowed) in each transmitted or received serial character.<br>• 00: 5-bit character (default)<br>• 01: 6-bit character<br>• 10: 7-bit character<br>• 11: 8-bit character |

## 53.4.5 MCR—Offset 4h

Modem Control Register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 4h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RSV | **MCR_RESERVED:** Reserved. |
| 5 | 0x0<br>RW | **MCR_AFE:** MCR-Auto Flow Enable (AFE):<br>This is a reserved bit in 16550 spec.<br>• 0: Auto-RTS and Auto-CTS are disabled.<br>• 1: Auto-CTS is enabled.<br>If MCR[RTS] is also set, both Auto-CTS and Auto-RTS is enabled.<br>Auto-RTS is not enabled if AFE is not set regardless of the state of MCR[RTS].<br>Autoflow automates the flow of data between the UART and the remote device. |
| 4 | 0x0<br>RW | **MCR_LOOP:** Modem Control Register-Loop back test mode (LOOP):<br>This bit provides a local loopback feature for diagnostic testing of the UART. This feature allows the processor to verify the UART transmit and receive data paths.<br>The Transmit, Receive, and Modem Control interrupts are operational in this mode.<br>• 0: Normal UART operation.<br>• 1: Loopback Test mode operation.<br>Serial output uart_txd pin is set to logic 1. Serial input uart_rxd pin is disconnected from external input line.<br>The output of the Transmitter Shift register is looped back into the Receiver Shift register.<br>The modem-control inputs (uart_cts, uart_dsr, uart_dcd, and uart_ri) are activated by MCR bits 0-3 instead of the modem-control inputs.<br>A Break signal can also be transferred from the transmitter section to the receiver section in Loopback mode. When LOOP is set to logic 1, the following will occur:<br>The TXD (transmitter output) pin is set to a logic-1 state.<br>The RXD (receiver input) pin is disconnected.<br>The output of the Transmitter Shift register is looped back into the Receiver-Shift register input.<br>The four modem-control inputs (uart_cts, uart_dsr, uart_dcd, and uart_ri) are disconnected from the pins and the modem-control output pins (uart_rts and uart_dtr) are forced to their inactive state.<br>The lower four bits of the Modem Control register (MCR) are connected to the upper four modem status register (MSR) bits.<br>Flow control can be tested; if autoflow is enabled, the MCR[RTS] bit has no effect on the nCTS input as nRTS is asserted by the autoflow logic.<br>• DTR = 1 forces DSR to a 1<br>• RTS = 1 forces CTS to a 1<br>• OUT1 = 1 forces RI to a 1<br>• OUT2= 1 forces DCD to a 1<br>Coming out of the Loopback Test mode may result in unpredictable activation of the delta bits (bits 3:0) in the Modem Status register (MSR).<br>It is recommended that MSR is read once to clear the delta bits in the MSR. |
| 3 | 0x0<br>RW | **MCR_OUT2:** Output2 (OUT2):<br>The function of the auxiliary output OUT2 bit differs depending on the mode of the UART.<br>• 0: uart_dcd UART pin input set low when in Loop-back mode<br>• 1: uart_dcd UART pin input set high when in Loop-back mode. |
| 2 | 0x0<br>RW | **MCR_OUT1:** Modem Control Register-output1 (OUT1):<br>The auxiliary output OUT1 bit is only used in Loopback Test mode.<br>• 0: uart_ri UART pin input set low when in Loop-back mode<br>• 1: uart_ri UART pin input set high when in Loop-back mode |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **MCR_RTS:**<br>When Auto Flow is disabled MCR[AFE] = 0:<br>• 0: Primary output uart_rts pin is forced to logic 1.<br>• 1: Primary output uart_rts pin is forced to logic 0.<br>When Auto Flow is enabled: MCR[AFE] = 1:<br>• 0: Auto-RTS is disabled.<br>• 1: Auto-RTS is enabled. |
| 0 | 0x0<br>RW | **MCR_DTR:** Modem Control Register-Data Terminal Ready (DTR):<br>This bit controls the Data Terminal Ready output.<br>• 0: Primary output uart_dtr pin is forced to logic 1.<br>• 1: Primary output uart_dtr pin is forced to logic 0.<br>The uart_dtr output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding modem or data set. |

## 53.4.6 LSR—Offset 5h

Line Status Register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 5h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **LSR_FIFOE:** Line Status Register - FIFO Error Status (FIFOE):<br>• 0: No character has error inside the Receive FIFO.<br>• 1: At least one character in Receiver FIFO has errors.<br>In non-FIFO mode, this bit is always set to 0. In FIFO mode, FIFOE is set to 1 when there is at least one parity error, framing error, or break indication for any of the characters in the FIFO.<br>A processor read to the Line Status register does not reset this bit.<br>FIFOE is reset when all error bytes have been read from the FIFO.<br>If DMA requests are enabled and FIFOE is set to 1, the Error Interrupt is generated and no Receive DMA request is generated even if the Receive FIFO reaches the trigger level.<br>If DMA requests are not enabled setting FIFOE to 1 does not generate an interrupt. |
| 6 | 0x0<br>RO/V | **LSR_TEMT:** Line Status Register - Transmitter Empty (TEMT):<br>• 0: There is data in the Transmit Shift register, the Transmit Holding register, or the FIFO (in FIFO mode).<br>• 1: All the data in the transmitter has been shifted out. |
| 5 | 0x0<br>RO/V | **LSR_TDRQ:** Line Status Register - Transmit data Request (TDRQ)<br>• 0: The UART is NOT ready to receive data for transmission.<br>• 1: The UART is ready to receive data for transmission.<br>The assertion of TDRQ causes the UART to interrupt the processor when the Transmit Data Request Interrupt Enable IER[TIE] is set high, and generates a DMA request to the DMA controller if DMA Request is enabled.<br>TDRQ is set to 1 when the new symbol is copied from Transmit Holding register into the Transmit Shift register.<br>The bit is reset to logic 0 concurrently with the loading of the Transmit Holding register by the processor.<br>In FIFO mode, TDRQ is set to logic 1 when the FIFO is less than half full.<br>It is cleared when the FIFO is more than half full. If more than 64 characters are loaded into the FIFO, the excess characters are lost. |
| 4 | 0x0<br>RO/C/V | **LSR_BI:** Line Status Register - Break Interrupt (BI):<br>• 0: No break signal has been received.<br>• 1: Break signal has been received.<br>BI is set to logic 1 when the received data input is held in the Spacing (logic 0) state for longer than a full character transmission time (that is, the total time of start bit + data bits + parity bit + stop bits).<br>The Break Indicator is reset when the processor reads the Line Status register.<br>In FIFO mode, only one Break character (equal to 0x00) is loaded into the FIFO regardless of the length of the Break condition.<br>BI shows the Break condition for the character at the bottom of the FIFO, not the most recent character received. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RO/C/V | **LSR_FE:** Line Status Register - frame error (FE):<br>• 0: No Framing error.<br>• 1: Framing error has occurred.<br>FE indicates that the received character did not have a valid stop bit. FE is set to logic 1 when the bit following the last data bit or parity bit is detected as logic 0 bit (spacing level).<br>If the Line Control register had been set for two stop bits, the receiver does not check for a valid second stop bit.<br>The FE indicator is reset when the processor reads the Line Status register.<br>The UART will resynchronize after a framing error by assuming that the framing error was due to the next start bit. Therefore it samples this start bit twice and then takes in the data.<br>In FIFO mode, FE shows a framing error for the character at the bottom of the FIFO, not for the most recent character received. |
| 2 | 0x0<br>RO/C/V | **LSR_PE:** Line Status Register - Parity Enable (PE):<br>• 0: No Parity error.<br>• 1: Parity error has occurred.<br>PE indicates that the received character does not have the correct even or odd parity, as selected by the Even Parity Select bit.<br>The PE is set to logic 1 upon detection of a parity error and is reset to logic 0 when the processor reads the Line Status register.<br>In FIFO mode, PE shows a parity error for the character at the bottom of the FIFO, not the most recent character received. |
| 1 | 0x0<br>RO/C/V | **LSR_OE:** Line Status Register - Overflow Error (OE):<br>• 0: No overflow error. Data has not been lost.<br>• 1: Overflow error. Receive data has been lost.<br>In non-FIFO mode, OE indicates that data in the Receiver Buffer register was not read by the processor before the next character was received; the new character is lost.<br>In FIFO mode, OE indicates that all 64 bytes of the FIFO are full and the most recently received byte has been discarded.<br>The OE indicator is set to logic 1 upon detection of an overflow condition and reset when the processor reads the Line Status register. |
| 0 | 0x0<br>RO/V | **LSR_DR:** Line Status Register - Data Ready (DR):<br>• 0: No data has been received.<br>• 1: Data is available in RBR or the FIFO.<br>DR is set to logic 1 when a complete incoming character has been received and transferred into the Receiver Buffer register or the FIFO.<br>In non-FIFO mode, DR is reset to 0 when the receive buffer is read.<br>In FIFO mode, DR is reset to logic 0 if the FIFO is empty (last character has been read from RBR) or the RESETRF bit is set in FCR. |

## 53.4.7 MSR—Offset 6h

Modem Status Register

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 6h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **MSR_DCD:** Modem Status Register - data Carrier Detect (DCD):<br>Loopback is disabled, MCR[LOOP] = 0<br>• 0: Primary input uart_dcd is logic 1.<br>• 1: Primary input uart_dcd is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1<br>• 0: MCR[OUT2] is logic 0.<br>• 1: MCR[OUT2] is logic 1. |
| 6 | 0x0<br>RO/V | **MSR_RI:** Modem Status Register - Ring Indicator (RI):<br>Loopback is disabled, MCR[LOOP] = 0<br>• 0: Primary input uart_ri is logic 1.<br>• 1: Primary input uart_ri is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1<br>• 0: MCR[OUT1] is logic 0.<br>• 1: MCR[OUT1] is logic 1. |
| 5 | 0x0<br>RO/V | **MSR_DSR:** Modem Status Register - Data Set Ready (DSR):<br>Loopback is disabled, MCR[LOOP] = 0<br>• 0: Primary input uart_dsr is logic 1.<br>• 1: Primary input uart_dsr is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1<br>• 0: MCR[DTR] is logic 0.<br>• 1: MCR[DTR] is logic 1. |
| 4 | 0x0<br>RO/V | **MSR_CTS:** Modem Status Register - Clear to Send (CTS):<br>Loopback is disabled, MCR[LOOP] = 0<br>• 0: Primary input uart_cts is logic 1.<br>• 1: Primary input uart_cts is logic 0.<br>Loopback is enabled, MCR[LOOP] = 1<br>• 0: MCR[RTS] is logic 0.<br>• 1: MCR[RTS] is logic 1. |
| 3 | 0x0<br>RO/C/V | **MSR_DDCR:** Modem Status Register - Delta Data Carrier Ready (DDCR):<br>• 0: No change in uart_dcr pin since last read of MSR<br>• 1: uart_dcr pin has changed state.<br>When DDCR is set the modem status interrupt will be generated if enabled in IER. |
| 2 | 0x0<br>RO/C/V | **MSR_TERI:** Modem Status Register - Trailing Edge Ring Indicator (TERI):<br>• 0: uart_ri pin has not changed from 0 to 1 since last read of MSR.<br>• 1: uart_ri pin has changed from 0 to 1.<br>When TERI is set the modem status interrupt will be generated if enabled in IER. |
| 1 | 0x0<br>RO/C/V | **MSR_DDSR:** Modem Status Register - Delta Data Set Ready (DDSR):<br>• 0: No change in uart_dsr pin since last read of MSR.<br>• 1: uart_dsr pin has changed state.<br>When DDSR is set the modem status interrupt will be generated if enabled in IER. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **MSR_DCTS:** Modem Status Register - Delta Clear to Send (DCTS):<br>• 0: No change in uart_cts pin since last read of MSR.<br>• 1: uart_cts pin has change state.<br>When DCTS is set the modem status interrupt will be generated if enabled in IER. |

## 53.4.8 SPR—Offset 7h

Scratch Pad Status Register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IOBA] + 7h

**IOBA Type:** PCI Configuration Register (Size: 32 bits)
**IOBA Reference:** [B:0, D:26, F:0,1,2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **SPR:** Scratch Pad Register (SCRATCH):<br>Writing to this field does not affect the operation of the UART in any way. |

§ §

# 54 Innovation Engine - B0, D27, F(0, 1, 3, 4)

## 54.1 Introduction and Index

The host-accessible registers for the Innovation Engine are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 54.1.1 Host Configuration Space—HECI1 (F:0)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 27 (decimal), Function 0. The offset addresses are listed.

**Table 54-1. Summary of PCI Configuration Registers HECI1—0/27/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E58086 | "Identifiers (HECI1_ID)—Offset 0h" on page 2454 |
| 4 | 2 | 0000 | "Command (HECI1_CMD)—Offset 4h" on page 2455 |
| 6 | 2 | 0010 | "Status (HECI1_STS)—Offset 6h" on page 2456 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI1_RID_CC)—Offset 8h" on page 2457 |
| C | 1 | 00 | "Cache Line Size (HECI1_CLS)—Offset Ch" on page 2457 |
| D | 1 | 00 | "Master Latency Timer (HECI1_MLT)—Offset Dh" on page 2457 |
| E | 1 | 80 | "Header Type (HECI1_HTYPE)—Offset Eh" on page 2458 |
| F | 1 | 00 | "Built In Self-Test (HECI1_BIST)—Offset Fh" on page 2458 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h" on page 2458 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h" on page 2459 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI1_SS)—Offset 2Ch" on page 2459 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI1_CAP)—Offset 34h" on page 2459 |
| 3C | 2 | 0100 | "Interrupt Information (HECI1_INTR)—Offset 3Ch" on page 2460 |
| 3E | 1 | 00 | "Minimum Grant (HECI1_MGNT)—Offset 3Eh" on page 2460 |
| 3F | 1 | 00 | "Maximum Latency (HECI1_MLAT)—Offset 3Fh" on page 2460 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI1_HFS)—Offset 40h" on page 2461 |
| 44 | 4 | 80000000 | "Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h" on page 2463 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h" on page 2464 |
| 4C | 4 | 00000000 | "Host General Status (HECI1_H_GS1)—Offset 4Ch" on page 2465 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI1_PID)—Offset 50h" on page 2465 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI1_PC)—Offset 52h" on page 2465 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h" on page 2466 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h" on page 2466 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h" on page 2467 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h" on page 2467 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch" on page 2467 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI1_H_GS2)—Offset 70h" on page 2467 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI1_H_GS3)—Offset 74h" on page 2468 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch" on page 2468 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh" on page 2468 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h" on page 2469 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h" on page 2469 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h" on page 2469 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h" on page 2470 |
| BC | 4 | 40000000 | "Host Extend Register Status (HECI1_HERS)—Offset BCh" on page 2470 |
| C0 | 4 | 00000000 | "Host Extend Register DW1. (HECI1_HER1)—Offset C0h" on page 2471 |

## Table 54-1. Summary of PCI Configuration Registers HECI1—0/27/0 (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| C4 | 4 | 00000000 | "Host Extend Register DW2. (HECI1_HER2)—Offset C4h" on page 2471 |
| C8 | 4 | 00000000 | "Host Extend Register DW3. (HECI1_HER3)—Offset C8h" on page 2471 |
| CC | 4 | 00000000 | "Host Extend Register DW4. (HECI1_HER4)—Offset CCh" on page 2471 |
| D0 | 4 | 00000000 | "Host Extend Register DW5. (HECI1_HER5)—Offset D0h" on page 2472 |
| D4 | 4 | 00000000 | "Host Extend Register DW6. (HECI1_HER6)—Offset D4h" on page 2472 |
| D8 | 4 | 00000000 | "Host Extend Register DW7. (HECI1_HER7)—Offset D8h" on page 2472 |
| DC | 4 | 00000000 | "Host Extend Register DW8. (HECI1_HER8)—Offset DCh" on page 2472 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI1_MANID)—Offset F8h" on page 2473 |

### 54.1.2 Host Configuration Space—HECI2 (F:1)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 27 (decimal), Function 1. The offset addresses are listed.

**Table 54-2. Summary of PCI Configuration Registers HECI2—0/27/1**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E68086 | "Identifiers (HECI2_ID)—Offset 0h" on page 2474 |
| 4 | 2 | 0000 | "Command (HECI2_CMD)—Offset 4h" on page 2475 |
| 6 | 2 | 0010 | "Status (HECI2_STS)—Offset 6h" on page 2476 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI2_RID_CC)—Offset 8h" on page 2477 |
| C | 1 | 00 | "Cache Line Size (HECI2_CLS)—Offset Ch" on page 2477 |
| D | 1 | 00 | "Master Latency Timer (HECI2_MLT)—Offset Dh" on page 2477 |
| E | 1 | 80 | "Header Type (HECI2_HTYPE)—Offset Eh" on page 2478 |
| F | 1 | 00 | "Built In Self-Test (HECI2_BIST)—Offset Fh" on page 2478 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h" on page 2478 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h" on page 2479 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI2_SS)—Offset 2Ch" on page 2479 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI2_CAP)—Offset 34h" on page 2479 |
| 3C | 2 | 0100 | "Interrupt Information (HECI2_INTR)—Offset 3Ch" on page 2480 |
| 3E | 1 | 00 | "Minimum Grant (HECI2_MGNT)—Offset 3Eh" on page 2480 |
| 3F | 1 | 00 | "Maximum Latency (HECI2_MLAT)—Offset 3Fh" on page 2480 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI2_HFS)—Offset 40h" on page 2481 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h" on page 2483 |
| 4C | 4 | 00000000 | "Host General Status (HECI2_H_GS1)—Offset 4Ch" on page 2484 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI2_PID)—Offset 50h" on page 2484 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI2_PC)—Offset 52h" on page 2485 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h" on page 2486 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h" on page 2486 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h" on page 2487 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h" on page 2487 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch" on page 2487 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI2_H_GS2)—Offset 70h" on page 2487 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI2_H_GS3)—Offset 74h" on page 2488 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch" on page 2488 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh" on page 2488 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h" on page 2489 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h" on page 2489 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h" on page 2489 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h" on page 2490 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI2_MANID)—Offset F8h" on page 2490 |

### 54.1.3 Host Configuration Space—HECI3 (F:4)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 27 (decimal), Function 4. The offset addresses are listed.

**Table 54-3. Summary of PCI Configuration Registers HECI3—0/27/4**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E98086 | "Identifiers (HECI3_ID)—Offset 0h" on page 2491 |
| 4 | 2 | 0000 | "Command (HECI3_CMD)—Offset 4h" on page 2492 |
| 6 | 2 | 0010 | "Status (HECI3_STS)—Offset 6h" on page 2493 |
| 8 | 4 | 07800000 | "Revision ID and Class Code (HECI3_RID_CC)—Offset 8h" on page 2494 |
| C | 1 | 00 | "Cache Line Size (HECI3_CLS)—Offset Ch" on page 2494 |
| D | 1 | 00 | "Master Latency Timer (HECI3_MLT)—Offset Dh" on page 2494 |
| E | 1 | 80 | "Header Type (HECI3_HTYPE)—Offset Eh" on page 2495 |
| F | 1 | 00 | "Built In Self-Test (HECI3_BIST)—Offset Fh" on page 2495 |
| 10 | 4 | 00000004 | "HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h" on page 2495 |
| 14 | 4 | 00000000 | "HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h" on page 2496 |
| 2C | 4 | 00000000 | "Sub System Identifiers (HECI3_SS)—Offset 2Ch" on page 2496 |
| 34 | 1 | 50 | "Capabilities Pointer (HECI3_CAP)—Offset 34h" on page 2496 |
| 3C | 2 | 0100 | "Interrupt Information (HECI3_INTR)—Offset 3Ch" on page 2497 |
| 3E | 1 | 00 | "Minimum Grant (HECI3_MGNT)—Offset 3Eh" on page 2497 |
| 3F | 1 | 00 | "Maximum Latency (HECI3_MLAT)—Offset 3Fh" on page 2497 |
| 40 | 4 | 00000000 | "Host Firmware Status (HECI3_HFS)—Offset 40h" on page 2498 |
| 48 | 4 | 00000000 | "General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h" on page 2500 |
| 4C | 4 | 00000000 | "Host General Status (HECI3_H_GS1)—Offset 4Ch" on page 2501 |
| 50 | 2 | 8C01 | "PCI Power Management Capability ID (HECI3_PID)—Offset 50h" on page 2501 |
| 52 | 2 | 4003 | "PCI Power Management Capabilities (HECI3_PC)—Offset 52h" on page 2502 |
| 54 | 2 | 0008 | "PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h" on page 2503 |
| 60 | 4 | 00000000 | "General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h" on page 2503 |
| 64 | 4 | 00000000 | "General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h" on page 2504 |
| 68 | 4 | 00000000 | "General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h" on page 2504 |
| 6C | 4 | 00000000 | "General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch" on page 2504 |
| 70 | 4 | 00000000 | "Host General Status 2 (HECI3_H_GS2)—Offset 70h" on page 2504 |
| 74 | 4 | 00000000 | "Host General Status 3 (HECI3_H_GS3)—Offset 74h" on page 2505 |
| 8C | 2 | 0005 | "Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch" on page 2505 |
| 8E | 2 | 0080 | "Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh" on page 2505 |
| 90 | 4 | 00000000 | "Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h" on page 2506 |
| 94 | 4 | 00000000 | "Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h" on page 2506 |
| 98 | 2 | 0000 | "Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h" on page 2506 |
| A0 | 1 | 00 | "HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h" on page 2507 |
| F8 | 4 | 00000000 | "Manufacturer's ID (HECI3_MANID)—Offset F8h" on page 2507 |

### 54.1.4 Host Configuration Space—KT (F:3)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 27 (decimal), Function 3. The offset addresses are listed.

**Table 54-4. Summary of PCI Configuration Registers KT (Host)—0/27/3**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E88086 | "Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h" on page 2508 |
| 4 | 4 | 00B00000 | "Status and Command (KT_HOST_STS_CMD)—Offset 4h" on page 2509 |
| 8 | 4 | 07000200 | "Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h" on page 2511 |
| C | 4 | 00800000 | "BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch" on page 2512 |
| 10 | 4 | 00000001 | "KT IO BAR (KT_HOST_IOBAR)—Offset 10h" on page 2512 |
| 14 | 4 | 00000000 | "KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h" on page 2513 |
| 28 | 4 | 00000000 | "Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h" on page 2513 |
| 2C | 4 | 00008086 | "Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch" on page 2514 |
| 30 | 4 | 00000000 | "Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h" on page 2514 |
| 34 | 4 | 00000040 | "Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h" on page 2514 |
| 3C | 4 | 00000000 | "Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch" on page 2515 |
| 40 | 4 | 00805005 | "MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h" on page 2516 |
| 44 | 4 | 00000000 | "MSI Message Address (KT_HOST_MSIMA)—Offset 44h" on page 2516 |
| 48 | 4 | 00000000 | "MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h" on page 2517 |
| 4C | 4 | 00000000 | "MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch" on page 2517 |
| 50 | 4 | 00230001 | "Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h" on page 2518 |
| 54 | 4 | 00000008 | "Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h" on page 2519 |
| F8 | 4 | 00000000 | "Process Stepping Information (KT_HOST_PSI)—Offset F8h" on page 2520 |

### 54.1.5 Host Configuration Space—IDE (F:2)

These registers are discovered in Host Configuration Space starting at Bus 0, Device 27 (decimal), Function 2. The offset addresses are listed.

**Table 54-5. Summary of PCI Configuration Registers—0/27/2**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E78086 | "Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h" on page 2521 |
| 4 | 4 | 00B00000 | "Status and Command (IDE_HOST_STS_CMD)—Offset 4h" on page 2522 |
| 8 | 4 | 01018500 | "Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h" on page 2524 |
| C | 4 | 00800000 | "BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch" on page 2525 |
| 10 | 4 | 00000001 | "IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h" on page 2525 |
| 14 | 4 | 00000001 | "IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h" on page 2526 |
| 18 | 4 | 00000001 | "IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h" on page 2526 |
| 1C | 4 | 00000001 | "IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch" on page 2526 |
| 20 | 4 | 00000001 | "IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h" on page 2527 |
| 28 | 4 | 00000000 | "Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h" on page 2527 |
| 2C | 4 | 00008086 | "Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch" on page 2527 |
| 30 | 4 | 00000000 | "Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h" on page 2528 |
| 34 | 4 | 00000040 | "Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h" on page 2528 |
| 3C | 4 | 00000000 | "Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch" on page 2529 |
| 40 | 4 | 00805005 | "MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h" on page 2530 |
| 44 | 4 | 00000000 | "MSI Message Address (IDE_HOST_MSIMA)—Offset 44h" on page 2530 |
| 48 | 4 | 00000000 | "MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h" on page 2531 |
| 4C | 4 | 00000000 | "MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch" on page 2531 |
| 50 | 4 | 00230001 | "Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h" on page 2532 |
| 54 | 4 | 00000008 | "Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h" on page 2533 |
| F8 | 4 | 00000000 | "Process Stepping Information (IDE_HOST_PSI)—Offset F8h" on page 2534 |

### 54.1.6 Host Memory Space—HECI1_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 54-6. Summary of Memory Mapped I/O Registers—HECI1_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI1_H_CB_WW)—Offset 0h" on page 2535 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI1_H_CSR)—Offset 4h" on page 2536 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h" on page 2537 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch" on page 2537 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI1_D0I3C)—Offset 800h" on page 2538 |

### 54.1.7 Host Memory Space—HECI2_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 54-7. Summary of Memory Mapped I/O Registers—HECI2_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI2_H_CB_WW)—Offset 0h" on page 2539 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI2_H_CSR)—Offset 4h" on page 2540 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h" on page 2541 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch" on page 2541 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI2_D0I3C)—Offset 800h" on page 2542 |

### 54.1.8 Host Memory Space—HECI3_MMIO_MBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 54-8. Summary of Memory Mapped I/O Registers—HECI3_MMIO_MBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Host CB Write Window (HECI3_H_CB_WW)—Offset 0h" on page 2543 |
| 4 | 4 | 80000000 | "Host Control and Status Register (HECI3_H_CSR)—Offset 4h" on page 2544 |
| 8 | 4 | FFFFFFFF | "CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h" on page 2545 |
| C | 4 | 80000000 | "CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch" on page 2545 |
| 800 | 4 | 00000000 | "D0i3 Control (HECI3_D0I3C)—Offset 800h" on page 2546 |

### 54.1.9 Host Memory Space—KT_HOST_MEMBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 54-9. Summary of Memory Mapped I/O Registers—KT_HOST_MEMBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "KT Receive Buffer Register (MEM_KTRBR)—Offset 0h" on page 2547 |
| 0 | 1 | 00 | "KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h" on page 2547 |
| 0 | 1 | 00 | "KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h" on page 2547 |
| 1 | 1 | 00 | "KT Interrupt Enable register (MEM_KTIER)—Offset 1h" on page 2548 |
| 1 | 1 | 00 | "KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h" on page 2548 |
| 2 | 1 | 01 | "KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h" on page 2549 |
| 2 | 1 | 00 | "KT FIFO Control register (MEM_KTFCR)—Offset 2h" on page 2550 |
| 3 | 1 | 03 | "KT Line Control register (MEM_KTLCR)—Offset 3h" on page 2551 |
| 4 | 1 | 00 | "KT Modem Control register (MEM_KTMCR)—Offset 4h" on page 2552 |
| 5 | 1 | 60 | "KT Line Status register (MEM_KTLSR)—Offset 5h" on page 2553 |
| 6 | 1 | 00 | "KT Modem Status register (MEM_KTMSR)—Offset 6h" on page 2554 |
| 7 | 1 | 00 | "KT Scratch register (MEM_KTSCR)—Offset 7h" on page 2554 |

### 54.1.10 Host Memory Space—Fixed Address

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 54-10. Summary of Memory Mapped I/O Registers—Fix Addresses**

| Memory Address (hex) | Bytes (decimal) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| FED40000 | 4 | 00000000 | "FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h" on page 2555 |
| FED40004 | 4 | 00000000 | "FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h" on page 2555 |
| FED40008 | 4 | 00000000 | "FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h" on page 2556 |
| FED4000C | 4 | 00000000 | "FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch" on page 2556 |
| FED40010 | 4 | 00000000 | "FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h" on page 2556 |
| FED40030 | 4 | 00000000 | "FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h" on page 2557 |
| FED40034 | 4 | 00000000 | "FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h" on page 2557 |
| FED40038 | 4 | 00000000 | "FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h" on page 2557 |
| FED40040 | 4 | 00000000 | "Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h" on page 2558 |
| FED40044 | 4 | 00000000 | "Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h" on page 2558 |
| FED40048 | 4 | 00000000 | "Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h" on page 2558 |
| FED4004C | 4 | 00000000 | "Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch" on page 2559 |
| FED40050 | 4 | 00000000 | "Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h" on page 2559 |
| FED40054 | 4 | 00000000 | "Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h" on page 2559 |
| FED40058 | 4 | 00000000 | "Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h" on page 2560 |
| FED4005C | 4 | 00000000 | "Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch" on page 2560 |
| FED40060 | 4 | 00000000 | "Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h" on page 2560 |
| FED40064 | 4 | 00000000 | "Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h" on page 2560 |
| FED40068 | 4 | 00000000 | "Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h" on page 2561 |
| FED4006C | 4 | 00000000 | "Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch" on page 2561 |
| FED40070 | 4 x 4 | 00000000 | "FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h" on page 2561 |
| FED40080 | 4 x 992 | 00000000 | "FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h" on page 2561 |

### 54.1.11 Host I/O Space—KT_HOST_IOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 54-11. Summary of I/O Registers—KT_HOST_IOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "KT Receive Buffer Register (IO_KTRBR)—Offset 0h" on page 2562 |
| 0 | 1 | 00 | "KT Transmitter Holding Register (IO_KTTHR)—Offset 0h" on page 2562 |
| 0 | 1 | 00 | "KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h" on page 2563 |
| 1 | 1 | 00 | "KT Interrupt Enable register (IO_KTIER)—Offset 1h" on page 2563 |
| 1 | 1 | 00 | "KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h" on page 2564 |
| 2 | 1 | 01 | "KT Interrupt Identification Register (IO_KTIIR)—Offset 2h" on page 2564 |
| 2 | 1 | 00 | "KT FIFO Control register (IO_KTFCR)—Offset 2h" on page 2565 |
| 3 | 1 | 03 | "KT Line Control register (IO_KTLCR)—Offset 3h" on page 2566 |
| 4 | 1 | 00 | "KT Modem Control register (IO_KTMCR)—Offset 4h" on page 2567 |
| 5 | 1 | 60 | "KT Line Status register (IO_KTLSR)—Offset 5h" on page 2568 |
| 6 | 1 | 00 | "KT Modem Status register (IO_KTMSR)—Offset 6h" on page 2569 |
| 7 | 1 | 00 | "KT Scratch register (IO_KTSCR)—Offset 7h" on page 2569 |

### 54.1.12 Host I/O Space—IDE_HOST_PCMDIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 54-12. Summary of I/O Registers—IDE_HOST_PCMDIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 0000 | "IDE Data Register (IDEDATA)—Offset 0h" on page 2570 |
| 1 | 1 | 00 | "IDE Features Register (IDEFR)—Offset 1h" on page 2570 |
| 1 | 1 | 00 | "IDE Error Register DEV0 (IDEERD0)—Offset 1h" on page 2571 |
| 1 | 1 | 00 | "IDE Error Register DEV1 (IDEERD1)—Offset 1h" on page 2571 |
| 2 | 1 | 00 | "IDE Sector Count In Register (IDESCIR)—Offset 2h" on page 2572 |
| 2 | 1 | 00 | "IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h" on page 2572 |
| 2 | 1 | 00 | "IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h" on page 2572 |
| 3 | 1 | 00 | "IDE Sector Number In Register (IDESNIR)—Offset 3h" on page 2573 |
| 3 | 1 | 00 | "IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h" on page 2573 |
| 3 | 1 | 00 | "IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h" on page 2573 |
| 4 | 1 | 00 | "IDE Cylinder Low In Register (IDECLIR)—Offset 4h" on page 2574 |
| 4 | 1 | 00 | "IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h" on page 2574 |
| 4 | 1 | 00 | "IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h" on page 2575 |
| 5 | 1 | 00 | "IDE Cylinder High In Register (IDECHIR)—Offset 5h" on page 2575 |
| 5 | 1 | 00 | "IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h" on page 2576 |
| 5 | 1 | 00 | "IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h" on page 2576 |
| 6 | 1 | 00 | "IDE Drive/Head In Register (IDEDHIR)—Offset 6h" on page 2577 |
| 6 | 1 | 00 | "IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h" on page 2577 |
| 6 | 1 | 00 | "IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h" on page 2578 |
| 7 | 1 | 00 | "IDE Command Register (IDECR)—Offset 7h" on page 2578 |
| 7 | 1 | 80 | "IDE Status Register DEV0 (IDESR0)—Offset 7h" on page 2579 |
| 7 | 1 | 80 | "IDE Status Register DEV1 (IDESR1)—Offset 7h" on page 2580 |

### 54.1.13 Host I/O Space—IDE_HOST_PCTLIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 54-13. Summary of I/O Registers—IDE_HOST_PCTLIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2 | 1 | 00 | "IDE Device Control Register (IDDCR)—Offset 2h" on page 2581 |
| 2 | 1 | 00 | "IDE Alternate Status Register (IDASR)—Offset 2h" on page 2581 |

### 54.1.14 Host I/O Space—IDE_HOST_BMIOBAR

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 54-14. Summary of I/O Registers—IDE_HOST_BMIOBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h" on page 2582 |
| 1 | 1 | 00 | "IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h" on page 2583 |
| 2 | 1 | 80 | "IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h" on page 2584 |
| 3 | 1 | 00 | "IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h" on page 2586 |
| 4 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h" on page 2586 |
| 5 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h" on page 2586 |
| 6 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h" on page 2587 |
| 7 | 1 | 00 | "IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h" on page 2587 |
| 8 | 1 | 00 | "IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h" on page 2588 |
| 9 | 1 | 00 | "IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h" on page 2588 |
| A | 1 | 80 | "IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah" on page 2589 |
| B | 1 | 00 | "IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh" on page 2590 |
| C | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch" on page 2590 |
| D | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh" on page 2591 |
| E | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh" on page 2591 |
| F | 1 | 00 | "IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh" on page 2592 |

### 54.1.15 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 54-15. Summary of Sideband Registers for KT (Host)—0xf4**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 0 | 00000000 | "KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h" on page 2593 |
| 4 | 00000000 | "BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h" on page 2593 |

**Table 54-16. Summary of Sideband Registers IDE (Host)—0xf4**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 0 | 00000000 | "IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h" on page 2594 |
| 4 | 00000000 | "BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h" on page 2594 |

## 54.2 Registers in Configuration Space—HECI1

### 54.2.1 Identifiers (HECI1_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 0h

**Default:** 19E58086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19E5<br>RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates that Intel is the vendor, assigned by the PCI SIG. |

## 54.2.2 Command (HECI1_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operations. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's ability to act as a system memory master for data transfers.<br>When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU.<br>***Note:*** This bit does not block HECI accesses to CSE-UMA. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controller’s memory-mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

### 54.2.3 Status (HECI1_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>• This bit is a 1 when the INTx# is asserted.<br>• This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

## 54.2.4 Revision ID and Class Code (HECI1_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

## 54.2.5 Cache Line Size (HECI1_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

## 54.2.6 Master Latency Timer (HECI1_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

### 54.2.7 Header Type (HECI1_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

### 54.2.8 Built In Self-Test (HECI1_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved |

### 54.2.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

### 54.2.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

### 54.2.11 Sub System Identifiers (HECI1_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

### 54.2.12 Capabilities Pointer (HECI1_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 54.2.13 Interrupt Information (HECI1_INTR)—Offset 3Ch

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0.<br>Programmed by BIOS over IOSF SB through corresponding Private CR register. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 54.2.14 Minimum Grant (HECI1_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 54.2.15 Maximum Latency (HECI1_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register
(Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 54.2.16 Host Firmware Status (HECI1_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 54.2.17 Miscellaneous Shadow (HECI1_MISC_SHDW)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 44h

**Default:** 80000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RO | **Miscellaneous Shadow Valid (MSVLD):** This bit is hardwired to 1 to indicate that this HECI device implements the Miscellaneous Shadow register.<br>This bit can be used by host software that is bus/dev/function number agnostic (such as HECI operating system driver) to discover whether the Miscellaneous Shadow register is implemented or not. |
| 30:17 | 0x0<br>RO | **RSVD_30_17:** Reserved |
| 16 | 0x0<br>RO/V | **CSE UMA Size Valid (CUSZV):** This bit indicates that firmware has written the CUSZ field.<br>This field reflects the value of CSE CUBA.CUSZV. |
| 15:7 | 0x0<br>RO | **Reserved (RSVD_15_7):** Reserved. |
| 6:0 | 0x0<br>RO/V | **CSE UMA Size (CUSZ):** These bits reflect firmware's desired size of CSEUMA memory region.<br>It is configured by firmware prior to bring up core power and allowing BIOS to initialize memory.<br>The legal CSEUMA sizes and encodings for this field are:<br>• 0000000b (0MB, no CSEUMA region)<br>• 0000001b (1MB)<br>• 0000010b (2MB)<br>• 0000100b (4MB)<br>• 0001000b (8MB)<br>• 0010000b (16MB)<br>• 0100000b (32MB)<br>• 1000000b (64MB)<br>This field reflects the value of CSE CUBA.CUSZ. |

## 54.2.18 General Status Shadow 1 (HECI1_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 54.2.19 Host General Status (HECI1_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host.<br>This field is not used by hardware. |

## 54.2.20 PCI Power Management Capability ID (HECI1_PID)—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list.<br>This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 54.2.21 PCI Power Management Capabilities (HECI1_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 54.2.22 PCI Power Management Control and Status (HECI1_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it.<br>CSE cannot clear this bit. Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over the SoC Sideband to the SoC PMC based on the PMES bit.<br>When reset these messages are not sent to the PMC. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state<br>The D1 and D2 states are not supported for this HECI host controller. |

## 54.2.23 General Status Shadow 2 (HECI1_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 54.2.24 General Status Shadow 3 (HECI1_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 54.2.25 General Status Shadow 4 (HECI1_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 54.2.26 General Status Shadow 5 (HECI1_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 54.2.27 Host General Status 2 (HECI1_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware. |

## 54.2.28 Host General Status 3 (HECI1_H_GS3)—Offset 74h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 54.2.29 Message Signaled Interrupt Identifiers (HECI1_MID)—Offset 8Ch

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be additional capability pointers or, if zero, indicates that this is the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates Message Signaled Interrupts (MSI). |

## 54.2.30 Message Signaled Interrupt Message Control (HECI1_MC)—Offset 8Eh

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

### 54.2.31 Message Signaled Interrupt Message Address (HECI1_MA)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

### 54.2.32 Message Signaled Interrupt Upper Address (HECI1_MUA)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

### 54.2.33 Message Signaled Interrupt Message Data (HECI1_MD)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:0] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 54.2.34 HECI Interrupt Delivery Mode (HECI1_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:0] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt<br>• 01 - Generate SCI<br>• 10 - Generate SMI<br>This field may be locked by writing 1 to HIDM_L bit. |

## 54.2.35 Host Extend Register Status (HECI1_HERS)—Offset BCh

This register is used to communicate the CSE FW measurement status information to host.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + BCh

**Default:** 40000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Extend Register Valid (ERV):** Set by FW after all FW has been loaded and the measurement data has been stored in HERx registers.<br>• If ERA field is SHA-1, the result of the extend operation is in HER:5-1.<br>• If ERA field is SHA-256, the result is in HER:8-1. |
| 30 | 0x1<br>RO | **Extend Feature Present (EFP):** This bit is hardwired to 1 to allow driver software to easily detect the chipset supports the Extend Registers FW measurement feature. |
| 29:4 | 0x0<br>RO | **Reserved (RSVD_29_4):** Reserved. |
| 3:0 | 0x0<br>RO/V | **Extend Register Algorithm (ERA):** This field indicates the hash algorithm used in the FW measurement Extend operation.<br>Encodings are:<br>• 0x0: SHA-1<br>• 0x2: SHA-256<br>• Other values: reserved.<br>This field is set by FW with the used hash algorithm value when the ERV bit is set to 1.<br>This field is meaningless when the ERV bit is 0.<br>This field does NOT have any defined reset value. |

## 54.2.36 Host Extend Register DW1. (HECI1_HER1)—Offset C0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW1 (ERDW1):** 1st DWORD result of the extend operation.<br>***Note:*** Note: Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.37 Host Extend Register DW2. (HECI1_HER2)—Offset C4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW2 (ERDW2):** 2nd DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.38 Host Extend Register DW3. (HECI1_HER3)—Offset C8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW3 (ERDW3):** 3rd DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.39 Host Extend Register DW4. (HECI1_HER4)—Offset CCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW4 (ERDW4):** 4th DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.40 Host Extend Register DW5. (HECI1_HER5)—Offset D0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW5 (ERDW5):** 5th DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.41 Host Extend Register DW6. (HECI1_HER6)—Offset D4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW6 (ERDW6):** 6th DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.42 Host Extend Register DW7. (HECI1_HER7)—Offset D8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + D8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW7 (ERDW7):** 7th DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.43 Host Extend Register DW8. (HECI1_HER8)—Offset DCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + DCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Host Extend Register DW8 (ERDW8):** 8th DWORD result of the extend operation.<br>***Note:*** Extend Operation is HER[5:1] if using SHA-1. If using SHA-256 then Extend Operation is HER[8:1]. |

## 54.2.44 Manufacturer's ID (HECI1_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:0] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part.<br>This field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** This field is set by the SoC. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process.<br>The dot portion of the process is reflected in bits [27:24].<br>This field can be used by software to differentiate steppings when Revision ID may not change. |

## 54.3 Registers in Configuration Space—HECI2

### 54.3.1 Identifiers (HECI2_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 0h

**Default:** 19E68086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19E6<br>RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates that Intel is the vendor, assigned by the PCI SIG. |

## 54.3.2 Command (HECI2_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operations. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's ability to act as a system memory master for data transfers.<br>When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU.<br>***Note:*** This bit does not block HECI accesses to CSE-UMA |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controllers memory-mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

### 54.3.3 Status (HECI2_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>This bit is a 1 when the INTx# is asserted.<br>This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

### 54.3.4 Revision ID and Class Code (HECI2_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

### 54.3.5 Cache Line Size (HECI2_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

### 54.3.6 Master Latency Timer (HECI2_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

## 54.3.7 Header Type (HECI2_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

## 54.3.8 Built In Self-Test (HECI2_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved |

## 54.3.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

### 54.3.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

### 54.3.11 Sub System Identifiers (HECI2_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

### 54.3.12 Capabilities Pointer (HECI2_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 54.3.13 Interrupt Information (HECI2_INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 54.3.14 Minimum Grant (HECI2_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 54.3.15 Maximum Latency (HECI2_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 54.3.16 Host Firmware Status (HECI2_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 54.3.17 General Status Shadow 1 (HECI2_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 54.3.18 Host General Status (HECI2_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host.<br>This field is not used by hardware. |

## 54.3.19 PCI Power Management Capability ID (HECI2_PID)—Offset 50h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list.<br>This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 54.3.20 PCI Power Management Capabilities (HECI2_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 54.3.21 PCI Power Management Control and Status (HECI2_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it.<br>CSE cannot clear this bit.<br>Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over the Soc Sideband to the SoC PMC based on the PMES bit.<br>When reset these messages are not sent to the PMC. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state<br>The D1 and D2 states are not supported for this HECI host controller. |

## 54.3.22 General Status Shadow 2 (HECI2_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 54.3.23 General Status Shadow 3 (HECI2_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 54.3.24 General Status Shadow 4 (HECI2_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 54.3.25 General Status Shadow 5 (HECI2_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 54.3.26 Host General Status 2 (HECI2_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host.<br>This field is not used by hardware. |

## 54.3.27 Host General Status 3 (HECI2_H_GS3)—Offset 74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 54.3.28 Message Signaled Interrupt Identifiers (HECI2_MID)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be additional capability pointers or, if zero, indicates that this is the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates Message Signaled Interrupts (MSI). |

## 54.3.29 Message Signaled Interrupt Message Control (HECI2_MC)—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

## 54.3.30 Message Signaled Interrupt Message Address (HECI2_MA)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 54.3.31 Message Signaled Interrupt Upper Address (HECI2_MUA)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

## 54.3.32 Message Signaled Interrupt Message Data (HECI2_MD)—Offset 98h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:1] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 54.3.33 HECI Interrupt Delivery Mode (HECI2_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:1] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt<br>• 01 - Generate SCI<br>• 10 - Generate SMI<br>This field may be locked by writing 1 to HIDM_L bit. |

## 54.3.34 Manufacturer's ID (HECI2_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:1] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part.<br>***Note:*** This field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** This field is set by the SoC. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process.<br>The dot portion of the process is reflected in bits [27:24].<br>This field can be used by software to differentiate steppings when Revision ID may not change. |

## 54.4 Registers in Configuration Space—HECI3

### 54.4.1 Identifiers (HECI3_ID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 0h

**Default:** 19E98086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19E9<br>RO/V | **Device ID (DID):** Indicates what device number assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** 16-bit field which indicates that Intel is the vendor, assigned by the PCI SIG. |

## 54.4.2 Command (HECI3_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD:** Reserved |
| 10 | 0x0<br>RW | **Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts.<br>This bit does not have any effect on MSI operations. |
| 9 | 0x0<br>RO | **Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0. |
| 8 | 0x0<br>RO | **SERR# Enable (SEE):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Wait Cycle Enable (WCC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **Parity Error Response Enable (PEE):** Not implemented, hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0 |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWIE):** Not implemented, hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not implemented, hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the HECI host controller's ability to act as a system memory master for data transfers.<br>When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition.<br>This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA.<br>When this bit is '0', HECI is blocked from generating MSI to the host CPU.<br>***Note:*** This bit does not block HECI accesses to CSE-UMA |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls access to the HECI host controllers memory-mapped register space. |
| 0 | 0x0<br>RO | **I/O Space Enable (IOSE):** Not implemented, hardwired to 0. |

## 54.4.3 Status (HECI3_STS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 6h

**Default:** 0010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented, hardwired to 0. |
| 14 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented, hardwired to 0. |
| 13 | 0x0<br>RO | **Received Master-Abort (RMA):** Not implemented, hardwired to 0. |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Not implemented, hardwired to 0. |
| 11 | 0x0<br>RO | **Signaled Target-Abort (STA):** Not implemented, hardwired to 0. |
| 10:9 | 0x0<br>RO | **DEVSEL# Timing (DEVT):** These bits are hardwired to 00. |
| 8 | 0x0<br>RO | **Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0. |
| 7 | 0x0<br>RO | **Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0. |
| 6 | 0x0<br>RO | **RSVD (RSVD_6_6):** Reserved |
| 5 | 0x0<br>RO | **66 MHz Capable (C66):** Not implemented, hardwired to 0. |
| 4 | 0x1<br>RO | **Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1. |
| 3 | 0x0<br>RO/V | **Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit.<br>This bit is a 1 when the INTx# is asserted.<br>This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register).<br>***Note:*** This bit is not set by an MSI. |
| 2:0 | 0x0<br>RO | **RSVD (RSVD_2_0):** Reserved |

## 54.4.4 Revision ID and Class Code (HECI3_RID_CC)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 8h

**Default:** 07800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Indicates the base class code of the HECI host controller device. |
| 23:16 | 0x80<br>RO | **Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device. |
| 15:8 | 0x0<br>RO | **Programming Interface (PI):** Indicates the programming interface of the HECI host controller device. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates stepping of the HECI host controller. |

## 54.4.5 Cache Line Size (HECI3_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented, hardwired to 0. |

## 54.4.6 Master Latency Timer (HECI3_MLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Master Latency Timer (MLT):** Not implemented, hardwired to 0. |

## 54.4.7 Header Type (HECI3_HTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + Eh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device. |
| 6:0 | 0x0<br>RO | **Header Layout (HL):** Indicates that the HECI host controller uses a target device layout. |

## 54.4.8 Built In Self-Test (HECI3_BIST)—Offset Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **BIST Capable (BC):** Not implemented, hardwired to 0. |
| 6:0 | 0x0<br>RO | **RSVD:** Reserved |

## 54.4.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space. |
| 11:4 | 0x0<br>RO | **Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PF):** Indicates that this range is not pre-fetchable. |
| 2:1 | 0x2<br>RO | **Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space. |
| 0 | 0x0<br>RO | **Resource Type Indicator (RTE):** Indicates a request for register memory space. |

## 54.4.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address High (BA_HI):** Upper 32 bits of base address of register memory space. |

## 54.4.11 Sub System Identifiers (HECI3_SS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Indicates the sub- system identifier.<br>This field should be programmed by BIOS during boot-up.<br>Once written, this register becomes Read Only. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Indicates the sub-system vendor identifier.<br>This field should be programmed by BIOS during boot- up.<br>Once written, this register becomes Read Only. |

## 54.4.12 Capabilities Pointer (HECI3_CAP)—Offset 34h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + 34h

**Default:** 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x50<br>RO | **Capability Pointer (CP):** Indicates the first capability pointer offset.<br>It points to the PCI power management capability offset. |

## 54.4.13 Interrupt Information (HECI3_INTR)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 3Ch

**Default:** 0100h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x1<br>RO/V | **Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses.<br>A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively.<br>When legacy interrupt is not used this register should be set to 0x0. |
| 7:0 | 0x0<br>RW | **Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to.<br>No hardware action is taken on this field. |

## 54.4.14 Minimum Grant (HECI3_MGNT)—Offset 3Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + 3Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Grant (GNT):** Not implemented, hardwired to 0. |

## 54.4.15 Maximum Latency (HECI3_MLAT)—Offset 3Fh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + 3Fh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Latency (LAT):** Not implemented, hardwired to 0. |

## 54.4.16 Host Firmware Status (HECI3_HFS)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **BIOS MSG ACK:** Acknowledge for register based BIOS message in MEI 1 H_GS Register. |
| 27:25 | 0x0<br>RO | **BIOS MSG ACK Data:** Message specific data for acknowledged BIOS message. |
| 24:20 | 0x0<br>RO | Reserved |
| 19:16 | 0x0<br>RO | **Operating Mode:** This field describes the current operating mode of Intel® ME.<br>• 1:0 - Reserved<br>• 2 - Debug Mode<br>• 14:3 - Reserved<br>• 15 - Intel® SPS firmware is running in Intel® ME |
| 15:12 | 0x0<br>RO | **Error Code:** If set to nonzero value the Intel® ME firmware has encountered a fatal error and stopped normal operation.<br>• 0 - No Error<br>• 1 - Uncategorized Failure - The Intel® ME firmware has experienced an uncategorized error. Further details of the failure can be found in the Extended Status Data.<br>• 2 - Disabled - Firmware was disabled on this platform.<br>• 3 - Image Failure - The Intel® ME firmware stored in the system flash is not valid. |
| 11 | 0x0<br>RO | **Update in Progress:** The bit is set if any type of Intel® ME firmware update is in progress. |
| 10 | 0x0<br>RO | **Recovery BUP Load Fault:** This bit is set when firmware is not able to load recovery bring-up from the flash. It means that the recovery section in Intel® ME region on the SPI flash. |
| 9 | 0x0<br>RO | **Init Complete:** When this but is not set firmware is still in initialization phase. When firmware has fully entered a stable state, this but is set to 1 and "Current State" field of this register provides the steady state of the Intel® ME subsystem. |
| 8:6 | 0x0<br>RO | **Operating State:** This field describes the current operating state of Intel® ME.<br>• 000 - Preboot<br>• 001 - M0 with UMA<br>• 010 - Reserved<br>• 011 - Reserved<br>• 100 - M3 without UMA<br>• 101 - M0 without UMA - normal state for Intel® SPS firmware<br>• 110 - Bring up<br>• 111 - M0 without UMA but with error |
| 5 | 0x0<br>RO | **FPT or Factory Defaults Bad:** This bit is set when the firmware discovers a bad checksum of Intel® ME region Flash Partition Table (FPT) or Factory Defaults. When this bit set, it may or may not have the error code shown in bit [15:12]. The system can get this bit clear only by re-flashing the whole Intel® ME region in the SPI Flash. |
| 4 | 0x0<br>RO | **Manufacturing Mode:** When this bit is set, the platform is still in manufacturing mode. Host can use this bit to inform user that the platform is NOT READY for production yet. This but is set as long as Intel® ME Region access is not locked for flash masters other than Intel® ME. For shipping machine, this bit MUST be 0. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:0 | 0x0<br>RO | **Current State:** This field describes the current operation state of the firmware.<br>• 0 - Reset - Intel® ME is in reset state, will exit this state within 1 millisecond<br>• 1 - Initialization - Intel® ME is initializing, will exit this state within 2 seconds<br>• 2 - Recovery - Intel® ME is in recovery mode, check ME1.GMES register to determine cause<br>• 3 - Reserved<br>• 4 - Disabled - Intel® ME functionality has been disabled, it executes idle loop<br>• 5 - Operational - Intel® ME is in normal operational state<br>• 6 - Reserved<br>• 7 - State Transition - Intel® ME sets this state before starting a transition to a new Operating State.<br>It is a temporary state, may appear on transition between Initialization and Operational. |

## 54.4.17 General Status Shadow 1 (HECI3_GS_SHDW1)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **EOP Status:** This bit presents the Intel® ME notion of EOP status.<br>If BIOS encounters this bit set to '1' during POST it signals an error in platform power flow. |
| 30:28 | 0x0<br>RO | **Infrastructure Progress Code:** This field identifies the infrastructure progress code:<br>• 0 - ROM - Intel® ME is in ROM phase<br>• 1 - BUP - Intel® ME is in BRINGUP phase<br>• 2 - uKernel - Intel® ME is in Micro Kernel phase<br>• 3 - Policy Module - Intel® ME is in Policy Module phase<br>• 4 - Other Module - Intel® ME is loading modules in M0 or M3 Operating State |
| 27:16 | 0x0<br>RO | **Extended Status Data:** These bits provide extended status data for the current state of operating of the firmware. Or, if the firmware is in a fatal error state, these bits provide extended error code information. |
| 15:13 | 0x0<br>RO | **Firmware Heartbeat:** This number increments approximately every second if Intel® ME firmware is running. |
| 12 | 0x0<br>RO | Intel® ME Target Image Boot Fault<br>• 0 - Target image loaded successfully<br>• 1 - Target image boot failed, switched to backup image or recovery image |
| 11:8 | 0x0<br>RO | Reserved |
| 7 | 0x0<br>RO | **Warm Reset Request:** If this bit is set, Intel® ME informs BIOS that a warm reset is requested by Intel® ME. |
| 6 | 0x0<br>RO | **MFS Failure:** If this bit is set, Intel® ME informs BIOS that Intel® ME File System failure has been detected during recent Intel® ME boot. |
| 5:4 | 0x0<br>RO | Reserved |
| 3:1 | 0x0<br>RO | **Recovery Cause:** If ME1.HFS.Current State indicates that Intel® ME firmware is running in recovery mode these bits provide the cause of this mode:<br>• 0 - Intel® ME recovery jumper asserted<br>• 1 - Security strap override jumper asserted<br>• 2 - Recovery forced with IPMI command<br>• 3 - Invalid flash configuration, either:<br>  • flash master access permissions configuration is wrong<br>  • VSCC entry is missing or wrong<br>  • flash erase block size in Intel® ME region configuration<br>• 4 - Intel® ME internal error<br>  Intel® ME could not start in operational mode because of some firmware problems.<br>• 5..7 - Reserved for future extensions |
| 0 | 0x0<br>RO | **BIST in Progress:** If this bit is set Intel® ME Built In Self-Test is in Progress. |

## 54.4.18 Host General Status (HECI3_H_GS1)—Offset 4Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status (H_GS1):** General status of Host.<br>This field is not used by hardware. |

## 54.4.19 PCI Power Management Capability ID (HECI3_PID)—Offset 50h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 50h

**Default:** 8C01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x8c<br>RO | **Next Capability (NEXT):** Indicates the location of the next capability item in the list.<br>This is the Message Signaled Interrupts capability. |
| 7:0 | 0x1<br>RO | **Cap ID (CID):** Indicates that this pointer is a PCI power management. |

## 54.4.20 PCI Power Management Capabilities (HECI3_PC)—Offset 52h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 52h

**Default:** 4003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x8<br>RO | **PME Support (PSUP):** Indicates the states that can generate PME#.<br>HECI can assert PME# from D3hot only. |
| 10 | 0x0<br>RO | **D2 Support (D2S):** The D2 state is not supported for the HECI host controller. |
| 9 | 0x0<br>RO | **D _Support (D1S):** The D1 state is not supported for the HECI host controller. |
| 8:6 | 0x0<br>RO | **Aux_Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported. |
| 5 | 0x0<br>RO | **Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RO | **PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#. |
| 2:0 | 0x3<br>RO | **Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. |

## 54.4.21 PCI Power Management Control and Status (HECI3_PMCS)—Offset 54h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 54h

**Default:** 0008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES.<br>This bit is cleared by host CPU writing a '1' to it.<br>CSE cannot clear this bit.<br>Host CPU writes with value '0' have no effect on this bit. |
| 14:9 | 0x0<br>RO | **Reserved (RSVD_14_9):** Reserved. |
| 8 | 0x0<br>RW | **PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over the SoC Sideband to the SoC PMC based on the PMES bit.<br>When reset these messages may not be sent. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform and internal reset. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are:<br>• 00 - D0 state<br>• 11 - D3HOT state<br>The D1 and D2 states are not supported for this HECI host controller. |

## 54.4.22 General Status Shadow 2 (HECI3_GS_SHDW2)—Offset 60h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 60h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2). |

## 54.4.23 General Status Shadow 3 (HECI3_GS_SHDW3)—Offset 64h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 64h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3). |

## 54.4.24 General Status Shadow 4 (HECI3_GS_SHDW4)—Offset 68h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 68h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4). |

## 54.4.25 General Status Shadow 5 (HECI3_GS_SHDW5)—Offset 6Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 6Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5). |

## 54.4.26 Host General Status 2 (HECI3_H_GS2)—Offset 70h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 70h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware. |

## 54.4.27 Host General Status 3 (HECI3_H_GS3)—Offset 74h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 74h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Host General Status 3 (H_GS3):** General status of Host.<br>This field is not used by hardware. |

## 54.4.28 Message Signaled Interrupt Identifiers (HECI3_MID)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 8Ch

**Default:** 0005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Next Pointer (NEXT):** Indicates the next item in the list.<br>This can be additional capability pointers or, if zero, indicates that this is the last item in the list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Indicates a Message Signaled Interrupt (MSI). |

## 54.4.29 Message Signaled Interrupt Message Control (HECI3_MC)—Offset 8Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 8Eh

**Default:** 0080h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7 | 0x1<br>RO | **64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. |
| 6:4 | 0x0<br>RO | **Multiple Message Enable (MME):** Not implemented, hardwired to 0. |
| 3:1 | 0x0<br>RO | **Multiple Message Capable (MMC):** Not implemented, hardwired to 0. |
| 0 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts. |

## 54.4.30 Message Signaled Interrupt Message Address (HECI3_MA)—Offset 90h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Address (ADDR):** Lower 32 bits of the system specified message address, always DW aligned. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 54.4.31 Message Signaled Interrupt Upper Address (HECI3_MUA)—Offset 94h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Upper Address (UADDR):** Upper 32 bits of the system specified message address.<br>This register is optional and only implemented if MC.C64=1. |

## 54.4.32 Message Signaled Interrupt Message Data (HECI3_MD)—Offset 98h

**Type:** PCI Configuration Register
(Size: 16 bits)

**Offset:** [B:0, D:27, F:4] + 98h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **Data (DATA):** This 16-bit field is programmed by system software if MSI is enabled.<br>Its content is driven onto the FSB during the data phase of the MSI memory write transaction. |

## 54.4.33 HECI Interrupt Delivery Mode (HECI3_HIDM)—Offset A0h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:27, F:4] + A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 2 | 0x0<br>RW/1S | **HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field. |
| 1:0 | 0x0<br>RW/L | **HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit.<br>They are interpreted as follows:<br>• 00 - Generate Legacy or MSI interrupt<br>• 01 - Generate SCI<br>• 10 - Generate SMI<br>This field may be locked by writing 1 to HIDM_L bit. |

## 54.4.34 Manufacturer's ID (HECI3_MANID)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:4] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD_31_28):** Reserved. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Indicates the dot process. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part.<br>This field can be used by software to differentiate steppings when Revision ID may not change. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** This field is set by the SoC. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Indicates the process.<br>The dot portion of the process is reflected in bits [27:24].<br>This field can be used by software to differentiate steppings when Revision ID may not change. |

## 54.5 Registers in Configuration Space—KT (Host)

### 54.5.1 Device ID and Vendor ID (KT_HOST_DID_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 0h

**Default:** 19E88086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19E8<br>RO/V | **Device ID (DID):** This field identifies the particular device. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies the manufacturer of the device.<br>The value of 0x8086 indicates Intel. |

## 54.5.2 Status and Command (KT_HOST_STS_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 4h

**Default:** 00B00000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented.<br>Hardwired to 0. |
| 30 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented.<br>Hardwired to 0. |
| 29 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit must be set by a master device whenever its transaction (except for Special Cycle) is completed with Unsupported Request Completion Status (a.k.a. Master-Abort).<br>All master devices must implement this bit. |
| 28 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit must be set by a master device whenever its transaction is completed with Completer Abort Completion Status (a.k.a. Target-Abort).<br>All master devices must implement this bit. |
| 27 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit must be set by a target device whenever it completes a Posted or Non-Posted transaction with a Completer Abort (a.k.a. Target-Abort) error.<br>Devices that will never signal Completer Abort (a.k.a. Target-Abort) do not need to implement this bit. |
| 26:25 | 0x0<br>RO | **Devsel Timing (DEVT):** These bits encode the timing of DEVSEL#.<br>There are three allowable timings for assertion of DEVSEL# as described below:<br>• 00b: fast<br>• 01b: medium<br>• 10b: slow<br>• 11b: reserved<br>These bits are read-only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write.<br>Hardwired to 00b. |
| 24 | 0x0<br>RO | **Master Data Parity Error (MDPE):** Not implemented.<br>Hardwired to 0. |
| 23 | 0x1<br>RO | **Fast Back to Back Capable (FBTBC):** This bit indicates whether or not the target is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.<br>This bit can be set to 1 if the device can accept these transactions and must be set to 0 otherwise.<br>Hardwired to 1. |
| 22 | 0x0<br>RO | **Reserved (RSVD3):** Reserved (RSVD) |
| 21 | 0x1<br>RO | **66 Mhz Capable (MCAP):** This bit indicates whether or not this device is capable of running at 66 MHz.<br>A value of 0 indicates 33 MHz.<br>A value of 1 indicates that the device is 66 MHz capable.<br>Hardwired to 1. |
| 20 | 0x1<br>RO | **Capabilities List (CAPL):** This optional read-only bit indicates whether or not this device implements the pointer for a New Capabilities linked list at offset 34h.<br>A value of zero indicates that no New Capabilities linked list is available.<br>A value of one indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 19 | 0x0<br>RO | **Interrupt Status (INTS):** This read-only bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted.<br>Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.<br>Read-only and hardwired to 0 for a device that does NOT support pin-based interrupt. |
| 18:16 | 0x0<br>RO | **Reserved (RSVD2):** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD1):** |
| 10 | 0x0<br>RW | **Interrupt Disable (INTD):** This bit disables the device/function from asserting INTx#.<br>A value of 0 enables the assertion of its INTx# signal.<br>A value of 1 disables the assertion of its INTx# signal.<br>This bit's state after RST# is 0. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBTBEN):** Not implemented.<br>Hardwired to 0. |
| 8 | 0x0<br>RO | **System Error Enable (SERREN):** Not implemented.<br>Hardwired to 0. |
| 7 | 0x0<br>RO | **Reserved (RSVD0):** |
| 6 | 0x0<br>RO | **Parity Error Response (PERRR):** Not implemented.<br>Hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** Not implemented.<br>Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWRIEN):** Not implemented.<br>Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycles (SPCYC):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the ability of a PCI device to issue Memory and I/O Read/Write Requests.<br>When this bit is Set, the PCI device function is allowed to issue Memory or I/O Requests.<br>When this bit is Clear, the PCI device function is not allowed to issue any Memory or I/O Requests.<br>***Note:*** As MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well.<br>Requests other than Memory or I/O Requests are not controlled by this bit.<br>Default value of this bit is 0b.<br>This bit is hardwired to 0b if a PCI device function does not generate Memory or I/O Requests. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Controls a device's response to Memory Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to Memory Space accesses.<br>State after RST# is 0. |
| 0 | 0x0<br>RW | **IO Space Enable (IOSE):** Controls a device's response to I/O Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to I/O Space accesses.<br>State after RST# is 0. |

### 54.5.3 Class Code and Revision ID (KT_HOST_CC_RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 8h

**Default:** 07000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x7<br>RO | **Base Class Code (BCC):** Identifies the Base Class Code of an external 16550-compatible serial controller device driver. |
| 23:16 | 0x0<br>RO | **Sub-Class Code (SCC):** Identifies the Sub-Class Code of an external 16550-compatible serial controller device driver. |
| 15:8 | 0x2<br>RO | **Programming Interface (PI):** Identifies the Programming Interface of an external 16550-compatible serial controller device driver. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** This register specifies a device specific revision identifier.<br>The value is chosen by the vendor.<br>Zero is an acceptable value.<br>This field should be viewed as a vendor defined extension to the Device ID. |

## 54.5.4 BIST, Header Type, Latency Timer, and Cache Line Size (KT_HOST_BIST_HTYPE_LT_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Built In Self Test (BIST):** Not implemented. Hardwired to 0. |
| 23 | 0x1<br>RO/V | **Header Type 1 (HTYPE1):** This bit identifies whether or not the device contains multiple functions.<br>• If the bit is 0, then the device is single function.<br>• If the bit is 1, then the device has multiple functions. |
| 22:16 | 0x0<br>RO | **Header Type 0 (HTYPE0):** This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space).<br>• The encoding 00h specifies the non-bridge Configuration Space Header.<br>• The encoding 01h specifies the PCI-to-PCI bridge Configuration Space Header.<br>• The encoding 02h specifies the CardBus bridge Configuration Space Header.<br>• All other encodings are reserved.<br>Hardwired to 0 to identify the non-bridge Configuration Space Header. |
| 15:8 | 0x0<br>RO | **Latency Timer (LT):** Not implemented.<br>Hardwired to 0. |
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented.<br>Hardwired to 0. |

## 54.5.5 KT IO BAR (KT_HOST_IOBAR)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space. |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.5.6 KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region. |
| 11:4 | 0x0<br>RO | **Memory Size (MEMSIZE):** Hardwired to 0 to indicate 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PREFETCH):** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.<br>Hardwired to 0 to indicate the device's memory space as non-prefetchable. |
| 2:1 | 0x0<br>RO | **Type (TYP):** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space. |
| 0 | 0x0<br>RO | **Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR. |

## 54.5.7 Cardbus CIS Pointer (KT_HOST_CCP)—Offset 28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Cardbus CIS Pointer (CCP):** Not implemented.<br>Hardwired to 0. |

### 54.5.8 Subsystem ID and Subsystem Vendor ID (KT_HOST_SID_SVID)—Offset 2Ch

These registers are used to uniquely identify the add-in card or subsystem where the PCI device resides. They provide a mechanism for add-in card vendors to distinguish their add-in cards from one another even though the add-in cards may have the same PCI controller on them (and, therefore, the same Vendor ID and Device ID).

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 2Ch

**Default:** 00008086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SID):** Value of this field is vendor-specific.<br>This is written by BIOS. No hardware action will be taken on this value.<br>Implementation Note: The Write-Once lock must be implemented per field. SID should have its own lock. |
| 15:0 | 0x8086<br>RW/O | **Subsystem Vendor ID (SVID):** This field identifies the vendor of the add-in card or subsystem.<br>This is written by BIOS. No hardware action will be taken on this value.<br>Implementation Note: The Write-Once lock must be implemented per field. SVID should have its own lock. |

### 54.5.9 Expansion ROM Base Address (KT_HOST_XRBAR)—Offset 30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Expansion ROM Base Address (XRBAR):** Not implemented.<br>Hardwired to 0. |

### 54.5.10 Capabilities List Pointer (KT_HOST_CAPP)—Offset 34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 34h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** |
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list which is the MSI Capability. |

## 54.5.11 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (KT_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Maximum Latency (MAXL):** Not implemented.<br>Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Minimum Grant (MING):** Not implemented.<br>Hardwired to 0. |
| 15:8 | 0x0<br>RO/V | **Interrupt Pin (INTP):** This register specifies which interrupt pin KT uses in PCI interrupt mode.<br><br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved.<br>***Note:*** This field shadows the KTHIPINR.IPIN field in the PTIO Host private CR space which is configured by BIOS over IOSF SB. |
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information. POST software will write the routing information into this register as it initializes and configures the system. |

## 54.5.12 MSI Message Control, Next Pointer and Capability ID (KT_HOST_MSIMC_MSINP_MSICID)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 40h

**Default:** 00805005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD):** |
| 24 | 0x0<br>RO | **Per Vector Masking Capable (PVMC):** Not implemented.<br>Hardwired to 0 to indicate the function does NOT support MSI per-vector masking. |
| 23 | 0x1<br>RO | **64 bit address capable (XAC):** Hardwired to 1 to indicate the function is capable of sending a 64-bit message address. |
| 22:20 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW.<br>This field is RW for software compatibility, but only one interrupt vector is ever sent by the function. |
| 19:17 | 0x0<br>RO | **Multiple Message Capable (MMC):** Encoded number of interrupt vectors requested by a device.<br>Hardwired to 0 to indicate one requested interrupt vector. |
| 16 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI interrupt delivery is enabled whereas pin-based interrupt delivery SHALL be disabled.<br>If cleared, prior to returning the configuration write (that clears this field) completion, the function must send any pending MSI(s). |
| 15:8 | 0x50<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 50h to point to the PM Capability list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Hardwired to 05h to indicate the linked list item as the MSI Capability registers. |

## 54.5.13 MSI Message Address (KT_HOST_MSIMA)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** 32-bit DW-aligned MSI message address. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** |

## 54.5.14 MSI Message Upper Address (KT_HOST_MSIMUA)—Offset 48h

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Upper 32-bit of a 64-bit DW-aligned MSI message address. |

## 54.5.15 MSI Message Data (KT_HOST_MSIMD)—Offset 4Ch

**Type:** PCI Configuration Register
(Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** |
| 15:0 | 0x0<br>RW | **Message Data (MD):** MSI Message Data |

## 54.5.16 Power Management Capabilities, Next Pointer and Capability ID (KT_HOST_PMCAP_PMNP_PMCID)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 50h

**Default:** 00230001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#.<br>A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state.<br>• bit(27) X XXX1b - PME# can be asserted from D0<br>• bit(28) X XX1Xb - PME# can be asserted from D1<br>• bit(29) X X1XXb - PME# can be asserted from D2<br>• bit(30) X 1XXXb - PME# can be asserted from D3hot<br>• bit(31) 1 XXXXb - PME# can be asserted from D3cold |
| 26 | 0x0<br>RO | **D2 Support (D2S):** Hardwired to 0 to indicate that this device does not support D2. |
| 25 | 0x0<br>RO | **D1 Support (D1S):** Hardwired to 0 to indicate that this device does not support D1. |
| 24:22 | 0x0<br>RO | **Aux Current (AUXC):** Not implemented.<br>Hardwired to 0. |
| 21 | 0x1<br>RO | **Device Specific Initialization (DSI):** indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.<br>A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state.<br>Hardwired to 1 to indicate Device Specific Initialization is required. |
| 20 | 0x0<br>RO | **Reserved (RSVD):** |
| 19 | 0x0<br>RO | **PME Clock (PMECLK):** Not implemented.<br>Hardwired to 0. |
| 18:16 | 0x3<br>RO | **Version (VER):** Hardwired to value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec. |
| 15:8 | 0x0<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 0 to indicate no more linked list item. |
| 7:0 | 0x1<br>RO | **Capability ID (CID):** Hardwired to 01h to indicate the linked list item as the PCI Power Management registers. |

## 54.5.17 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (KT_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + 54h

**Default:** 00000008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Data:** Not implemented. Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Control/Status Register Bridge Support Extensions (CSRBSE):** Not implemented. Hardwired to 0. |
| 15 | 0x0<br>RO | **PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit.<br>• Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled).<br>• Writing a 0 has no effect.<br>If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.<br>Not implemented. Hardwired to 0. |
| 14:13 | 0x0<br>RO | **Data Scale (DS):** Not implemented. Hardwired to 0. |
| 12:9 | 0x0<br>RO | **Data Select (DSEL):** Not implemented. Hardwired to 0. |
| 8 | 0x0<br>RO | **PME Enable (PMEEN):** A 1 enables the function to assert PME. When 0, PME assertion is disabled. This bit defaults to 0 if the function does not support PME generation from D3cold.<br>If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.<br>Not implemented. Hardwired to 0. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br>Configuration Context is preserved.<br>Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. |
| 2 | 0x0<br>RO | **Reserved (RSVD0):** |
| 1:0 | 0x0<br>RW | **Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state.<br>The definition of the field values is given below:<br>• 00b - D0<br>• 01b - D1<br>• 10b - D2<br>• 11b - D3hot<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. |

## 54.5.18 Process Stepping Information (KT_HOST_PSI)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:3] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Not implemented. Hardwired to 0. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Reserved. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping (MSID):** Reserved. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** Reserved. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Reserved. |

## 54.6 Registers in Configuration Space—IDE (Host)

### 54.6.1 Device ID and Vendor ID (IDE_HOST_DID_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 0h

**Default:** 19E78086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19e7<br>RO/V | **Device ID (DID):** This field identifies the particular device.<br>This identifier is allocated by the vendor. |
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** This field identifies the manufacturer of the device. The value of 0x8086 indicates Intel. |

## 54.6.2 Status and Command (IDE_HOST_STS_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 4h

**Default:** 00B00000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Detected Parity Error (DPE):** Not implemented. Hardwired to 0. |
| 30 | 0x0<br>RO | **Signaled System Error (SSE):** Not implemented. Hardwired to 0. |
| 29 | 0x0<br>RW/1C/V | **Received Master Abort (RMA):** This bit must be set by a master device whenever its transaction (except for Special Cycle) is completed with Unsupported Request Completion Status (a.k.a. Master-Abort).<br>All master devices must implement this bit. |
| 28 | 0x0<br>RW/1C/V | **Received Target Abort (RTA):** This bit must be set by a master device whenever its transaction is completed with Completer Abort Completion Status (a.k.a. Target-Abort).<br>All master devices must implement this bit. |
| 27 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** This bit must be set by a target device whenever it completes a Posted or Non-Posted transaction with a Completer Abort (a.k.a. Target-Abort) error.<br>Devices that will never signal Completer Abort (a.k.a. Target-Abort) do not need to implement this bit. |
| 26:25 | 0x0<br>RO | **Devsel Timing (DEVT):** These bits encode the timing of DEVSEL#.<br>There are three allowable timings for assertion of DEVSEL# as described below:<br>• 00b: fast<br>• 01b: medium<br>• 10b: slow<br>• 11b: reserved<br>These bits are read-only and must indicate the slowest time that a device asserts DEVSEL# for any bus command except Configuration Read and Configuration Write.<br>Hardwired to 00b. |
| 24 | 0x0<br>RO | **Master Data Parity Error (MDPE):** Not implemented. Hardwired to 0. |
| 23 | 0x1<br>RO | **Fast Back to Back Capable (FBTBC):** This bit indicates whether or not the target is capable of accepting fast back-to-back transactions when the transactions are not to the same agent.<br>This bit can be set to 1 if the device can accept these transactions and must be set to 0 otherwise.<br>Hardwired to 1. |
| 22 | 0x0<br>RO | **Reserved (RSVD3):** Reserved (RSVD |
| 21 | 0x1<br>RO | **66 Mhz Capable (MCAP):** This bit indicates whether or not this device is capable of running at 66 MHz.<br>• A value of 0 indicates 33 MHz<br>• A value of 1 indicates that the device is 66 MHz capable<br>Hardwired to 1. |
| 20 | 0x1<br>RO | **Capabilities List (CAPL):** This optional read-only bit indicates whether or not this device implements the pointer for a New Capabilities linked list at offset 34h.<br>• A value of zero indicates that no New Capabilities linked list is available.<br>• A value of one indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities. |
| 19 | 0x0<br>RO | **Interrupt Status (INTS):** This read-only bit reflects the state of the interrupt in the device/function.<br>Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted.<br>Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.<br>Read-only and hardwired to 0 for a device that does NOT support pin-based interrupt. |
| 18:16 | 0x0<br>RO | **Reserved (RSVD2):** |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD1):** |
| 10 | 0x0<br>RW | **Interrupt Disable (INTD):** This bit disables the device/function from asserting INTx#.<br>• A value of 0 enables the assertion of its INTx# signal<br>• A value of 1 disables the assertion of its INTx# signal<br>This bit's state after RST# is 0. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBTBEN):** Not implemented. Hardwired to 0. |
| 8 | 0x0<br>RO | **System Error Enable (SERREN):** Not implemented. Hardwired to 0. |
| 7 | 0x0<br>RO | **Reserved (RSVD0):** |
| 6 | 0x0<br>RO | **Parity Error Response (PERRR):** Not implemented. Hardwired to 0. |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** Not implemented. Hardwired to 0. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWRIEN):** Not implemented. Hardwired to 0. |
| 3 | 0x0<br>RO | **Special Cycles (SPCYC):** Not implemented. Hardwired to 0. |
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Controls the ability of a PCI device to issue Memory and I/O Read/Write Requests.<br>When this bit is Set, the PCI device function is allowed to issue Memory or I/O Requests.<br>When this bit is Clear, the PCI device function is not allowed to issue any Memory or I/O Requests.<br>***Note:*** As MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit to 0b disables MSI/MSI-X interrupt Messages as well.<br>Requests other than Memory or I/O Requests are not controlled by this bit.<br>Default value of this bit is 0b.<br>This bit is hardwired to 0b if a PCI device function does not generate Memory or I/O Requests. |
| 1 | 0x0<br>RO | **Memory Space Enable (MSE):** Controls a device's response to Memory Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to Memory Space accesses.<br>State after RST# is 0.<br>Read-only and hardwired to 0 because IDE does NOT support Memory Space accesses. |
| 0 | 0x0<br>RW | **IO Space Enable (IOSE):** Controls a device's response to I/O Space accesses.<br>• A value of 0 disables the device response.<br>• A value of 1 allows the device to respond to I/O Space accesses.<br>State after RST# is 0. |

### 54.6.3 Class Code and Revision ID (IDE_HOST_CC_RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 8h

**Default:** 01018500h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x1<br>RO | **Base Class Code (BCC):** Identifies the Base Class Code of an external IDE controller device driver. |
| 23:16 | 0x1<br>RO | **Sub-Class Code (SCC):** Identifies the Sub-Class Code of an external IDE controller device driver. |
| 15:8 | 0x85<br>RO | **Programming Interface (PI):** Identifies the Programming Interface of an IDE controller device driver. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** This register specifies a device specific revision identifier.<br>The value is chosen by the vendor. Zero is an acceptable value.<br>This field should be viewed as a vendor defined extension to the Device ID. |

## 54.6.4 BIST, Header Type, Latency Timer, and Cache Line Size (IDE_HOST_BIST_HTYPE_LT_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + Ch

**Default:** 00800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Built In Self Test (BIST):** Not implemented. Hardwired to 0. |
| 23 | 0x1<br>RO/V | **Header Type 1 (HTYPE1):** This bit identifies whether or not the device contains multiple functions.<br>• If the bit is 0, then the device is single function<br>• If the bit is 1, then the device has multiple functions |
| 22:16 | 0x0<br>RO | **Header Type 0 (HTYPE0):** This field identifies the layout of the second part of the predefined header (beginning at byte 10h in Configuration Space).<br>• The encoding 00h specifies the non-bridge Configuration Space Header.<br>• The encoding 01h specifies the PCI-to-PCI bridge Configuration Space Header.<br>• The encoding 02h specifies the CardBus bridge Configuration Space Header.<br>• All other encodings are reserved.<br>Hardwired to 0 to identify the non-bridge Configuration Space Header. |
| 15:8 | 0x0<br>RO | **Latency Timer (LT):** Not implemented. Hardwired to 0. |
| 7:0 | 0x0<br>RO | **Cache Line Size (CLS):** Not implemented. Hardwired to 0. |

## 54.6.5 IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 10h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space. |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.6.6 IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 14h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0 RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 1 | 0x0 RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1 RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.6.7 IDE Secondary Command Block IO BAR (IDE_HOST_SCMDIOBAR)—Offset 18h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 18h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0 RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 2 | 0x0 RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 8B of IO space. |
| 1 | 0x0 RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1 RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.6.8 IDE Secondary Control Block IO BAR (IDE_HOST_SCTLIOBAR)—Offset 1Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 1Ch

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0 RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 1 | 0x0 RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1 RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.6.9 IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 20h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RW | **IO BAR (IOBAR):** Software programs this space with the base address of the device's IO region. |
| 3:2 | 0x0<br>RO | **IO Size (IOSIZE):** Hardwired to 0 to indicate 16B of IO space. |
| 1 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0 for an IO BAR. |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSPACE):** Hardwired to 1 to identify an IO BAR. |

## 54.6.10 Cardbus CIS Pointer (IDE_HOST_CCP)—Offset 28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Cardbus CIS Pointer (CCP):** Not implemented.<br>Hardwired to 0. |

## 54.6.11 Subsystem ID and Subsystem Vendor ID (IDE_HOST_SID_SVID)—Offset 2Ch

These registers are used to uniquely identify the add-in card or subsystem where the PCI device resides. They provide a mechanism for add-in card vendors to distinguish their add-in cards from one another even though the add-in cards may have the same PCI controller on them (and, therefore, the same Vendor ID and Device ID).

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 2Ch

**Default:** 00008086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SID):** Value of this field is vendor-specific.<br>This is written by BIOS. No hardware action will be taken on this value.<br>Implementation Note: The Write-Once lock must be implemented per field. SID should have its own lock. |
| 15:0 | 0x8086<br>RW/O | **Subsystem Vendor ID (SVID):** This field identifies the vendor of the add-in card or subsystem.<br>This is written by BIOS. No hardware action will be taken on this value.<br>Implementation Note: The Write-Once lock must be implemented per field. SVID should have its own lock. |

## 54.6.12 Expansion ROM Base Address (IDE_HOST_XRBAR)—Offset 30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Expansion ROM Base Address (XRBAR):** Not implemented. Hardwired to 0. |

## 54.6.13 Capabilities List Pointer (IDE_HOST_CAPP)—Offset 34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 34h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** |
| 7:0 | 0x40<br>RO | **Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list which is the MSI Capability. |

## 54.6.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (IDE_HOST_MAXL_MING_INTP_INTL)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Maximum Latency (MAXL):** Not implemented. Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Minimum Grant (MING):** Not implemented. Hardwired to 0. |
| 15:8 | 0x0<br>RO/V | **Interrupt Pin (INTP):** This register specifies which interrupt pin IDE uses in PCI interrupt mode.<br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved. |
| 7:0 | 0x0<br>RW | **Interrupt Line (INTL):** This register is used to communicate interrupt line routing information. POST software will write the routing information into this register as it initializes and configures the system. |

## 54.6.15 MSI Message Control, Next Pointer and Capability ID (IDE_HOST_MSIMC_MSINP_MSICID)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 40h

**Default:** 00805005h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD):** |
| 24 | 0x0<br>RO | **Per Vector Masking Capable (PVMC):** Not implemented. Hardwired to 0 to indicate the function does NOT support MSI per-vector masking. |
| 23 | 0x1<br>RO | **64 bit address capable (XAC):** Hardwired to 1 to indicate the function is capable of sending a 64-bit message address. |
| 22:20 | 0x0<br>RW | **Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW.<br>This field is RW for software compatibility, but only one interrupt vector is ever sent by the function. |
| 19:17 | 0x0<br>RO | **Multiple Message Capable (MMC):** Encoded number of interrupt vectors requested by a device.<br>Hardwired to 0 to indicate one requested interrupt vector. |
| 16 | 0x0<br>RW | **MSI Enable (MSIE):** If set, MSI interrupt delivery is enabled whereas pin-based interrupt delivery SHALL be disabled.<br>If cleared, prior to returning the configuration write (that clears this field) completion, the function must send any pending MSI(s). |
| 15:8 | 0x50<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 50h to point to the PM Capability list. |
| 7:0 | 0x5<br>RO | **Capability ID (CID):** Hardwired to 05h to indicate the linked list item as the MSI Capability registers. |

## 54.6.16 MSI Message Address (IDE_HOST_MSIMA)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RW | **Message Address (MA):** 32-bit DW-aligned MSI message address. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD):** |

## 54.6.17 MSI Message Upper Address (IDE_HOST_MSIMUA)—Offset 48h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Message Upper Address (MUA):** Upper 32-bit of a 64-bit DW-aligned MSI message address. |

## 54.6.18 MSI Message Data (IDE_HOST_MSIMD)—Offset 4Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** |
| 15:0 | 0x0<br>RW | **Message Data (MD):** MSI Message Data |

## 54.6.19 Power Management Capabilities, Next Pointer and Capability ID (IDE_HOST_PMCAP_PMNP_PMCID)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 50h

**Default:** 00230001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#.<br>A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state.<br>• bit(27) X XXX1b - PME# can be asserted from D0<br>• bit(28) X XX1Xb - PME# can be asserted from D1<br>• bit(29) X X1XXb - PME# can be asserted from D2<br>• bit(30) X 1XXXb - PME# can be asserted from D3hot<br>• bit(31) 1 XXXXb - PME# can be asserted from D3cold |
| 26 | 0x0<br>RO | **D2 Support (D2S):** Hardwired to 0 to indicate that this device does not support D2. |
| 25 | 0x0<br>RO | **D1 Support (D1S):** Hardwired to 0 to indicate that this device does not support D1. |
| 24:22 | 0x0<br>RO | **Aux Current (AUXC):** Not implemented.Hardwired to 0. |
| 21 | 0x1<br>RO | **Device Specific Initialization (DSI):** indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it.<br>A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state.<br>Hardwired to 1 to indicate Device Specific Initialization is required. |
| 20 | 0x0<br>RO | **Reserved (RSVD):** |
| 19 | 0x0<br>RO | **PME Clock (PMECLK):** Not implemented.Hardwired to 0. |
| 18:16 | 0x3<br>RO | **Version (VER):** Hardwired to value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec. |
| 15:8 | 0x0<br>RO | **Next Item Pointer (NP):** Indicates the pointer for the next entry in the capabilities list.<br>Hardwired to 0 to indicate no more linked list item. |
| 7:0 | 0x1<br>RO | **Capability ID (CID):** Hardwired to 01h to indicate the linked list item as the PCI Power Management registers. |

## 54.6.20 Power Management Data, Control/Status Register Bridge Support Extensions, Control and Status (IDE_HOST_PMD_PMCSRBSE_PMCSR)—Offset 54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + 54h

**Default:** 00000008h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Data:** Not implemented. Hardwired to 0. |
| 23:16 | 0x0<br>RO | **Control/Status Register Bridge Support Extensions (CSRBSE):** Not implemented. Hardwired to 0. |
| 15 | 0x0<br>RO | **PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit.<br>• Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled).<br>• Writing a 0 has no effect.<br>If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded.<br>Not implemented. Hardwired to 0. |
| 14:13 | 0x0<br>RO | **Data Scale (DS):** Not implemented. Hardwired to 0. |
| 12:9 | 0x0<br>RO | **Data Select (DSEL):** Not implemented. Hardwired to 0. |
| 8 | 0x0<br>RO | **PME Enable (PMEEN):** A 1 enables the function to assert PME.<br>When 0, PME assertion is disabled.<br>This bit defaults to 0 if the function does not support PME generation from D3cold.<br>If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.<br>Not implemented. Hardwired to 0. |
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** |
| 3 | 0x1<br>RO | **No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset.<br>Configuration Context is preserved.<br>Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. |
| 2 | 0x0<br>RO | **Reserved (RSVD0):** |
| 1:0 | 0x0<br>RW | **Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state.<br>The definition of the field values is given below:<br>• 00b - D0<br>• 01b - D1<br>• 10b - D2<br>• 11b - D3hot<br>If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. |

## 54.6.21 Process Stepping Information (IDE_HOST_PSI)—Offset F8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:27, F:2] + F8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Not implemented. Hardwired to 0. |
| 27:24 | 0x0<br>RO/V | **Dot Portion of Process ID (DPOP):** Reserved. |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping (MSID):** Reserved. |
| 15:8 | 0x0<br>RO/V | **Manufacturing ID (MID):** Reserved. |
| 7:0 | 0x0<br>RO/V | **Process Portion of Process ID (PPOP):** Reserved. |

## 54.7 Registers in Memory Space—HECI1_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI1_MMIO_MBAR_LO described in Section 54.2.9, "HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO)—Offset 10h" on page 2458 and HECI1_MMIO_MBAR_HI described in Section 54.2.10, "HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI)—Offset 14h" on page 2459.

### 54.7.1 Host CB Write Window (HECI1_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI1_MMIO_MBAR] + 0h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:27, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data.<br>Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 54.7.2 Host Control and Status Register (HECI1_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:27, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br>Programmers note: This field is implemented with a '1-hot' scheme. Only one bit will be set to a '1' at a time.<br>Each bit position represents the value n of a buffer depth of (2^n). For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA. HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 54.7.3 CSE Circular Buffer Read Window (HECI1_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:27, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 54.7.4 CSE Control and Status Register Host Access (HECI1_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:27, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 54.7.5 D0i3 Control (HECI1_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI1_MMIO_MBAR] + 800h

**Default:** 00000000h

**HECI1_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI1_MMIO_MBAR Reference:** [B:0, D:27, F:0] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 54.8 Registers in Memory Space—HECI2_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI2_MMIO_MBAR_LO described in Section 54.3.9, "HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO)—Offset 10h" on page 2478 and HECI2_MMIO_MBAR_HI described in Section 54.3.10, "HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI)—Offset 14h" on page 2479.

### 54.8.1 Host CB Write Window (HECI2_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI2_MMIO_MBAR] + 0h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:27, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data. Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 54.8.2 Host Control and Status Register (HECI2_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:27, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br><br>Programmers note: This field is implemented with a '1-hot' scheme. Only one bit will be set to a '1' at a time.<br>Each bit position represents the value n of a buffer depth of (2^n).<br>For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA. HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 54.8.3 CSE Circular Buffer Read Window (HECI2_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:27, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1. When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 54.8.4 CSE Control and Status Register Host Access (HECI2_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI2_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:27, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 54.8.5 D0i3 Control (HECI2_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI2_MMIO_MBAR] + 800h

**HECI2_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI2_MMIO_MBAR Reference:** [B:0, D:27, F:1] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 54.9 Registers in Memory Space—HECI3_MMIO_MBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) HECI3_MMIO_MBAR_LO described in Section 54.4.9, "HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO)—Offset 10h" on page 2495 and HECI3_MMIO_MBAR_HI described in Section 54.4.10, "HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI)—Offset 14h" on page 2496.

### 54.9.1 Host CB Write Window (HECI3_H_CB_WW)—Offset 0h

This register is for host to write into its Circular Buffer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI3_MMIO_MBAR] + 0h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:27, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **Host Circular Buffer Write Window Field (H_CB_WWF):** This field is for host to write into its circular buffer.<br>This field is write only, reads will return arbitrary data.<br>Writes to this register will increment the H_CBWP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, writes to this register have no effect and are not delivered to the H_CB, nor is H_CBWP incremented. |

## 54.9.2 Host Control and Status Register (HECI3_H_CSR)—Offset 4h

This register reports status information about the host circular buffer (H_CB) and allows host software to control interrupt generation.

Note to software: reserved bits in this register must be set to 0 whenever this register is written.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + 4h

**Default:** 80000000h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:27, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **Host Circular Buffer Depth (H_CBD):** This field indicates the maximum number of 32 bit entries available in the host circular buffer (H_CB).<br>Host software uses this field along with the H_CBRP and H_CBWP fields to calculate the number of valid entries in the H_CB to read or number of entries available for write.<br><br>Programmers note: This field is implemented with a '1-hot' scheme.<br>Only one bit will be set to a '1' at a time.<br>Each bit position represents the value n of a buffer depth of (2^n).<br>For example, when bit# 1 is 1, the buffer depth is 2; when bit#2 is 1, the buffer depth is 4, etc.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **Host CB Write Pointer (H_CBWP):** Points to next location in the H_CB for host to write the data.<br>Software uses this field along with H_CBRP and H_CBD fields to calculate the number of valid entries in the H_CB to read or number of entries available for write. |
| 15:8 | 0x0<br>RO/V | **Host CB Read Pointer (H_CBRP):** Points to next location in the H_CB where a valid data is available for embedded controller to read.<br>Software uses this field along with H_CBWR and H_CBD fields to calculate the number of valid entries in the host CB to read or number of entries available for write. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Host D0I3 Interrupt Status (H_D0I3C_IS):** HW sets this bit to 1 when D0I3C.H_D0I3_IR is set and D0I3C.H_D0I3_CIP transitions from 1 to 0.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_D0I3C_IE has no effect on this bit. |
| 5 | 0x0<br>RW | **Host D0I3 Interrupt Enable (H_D0I3C_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_D0I3C_IS is set to 1. |
| 4 | 0x0<br>RW | **Host Reset (H_RST):** Setting this bit to 1 will initiate the HECI reset sequence to get the circular buffers into a known good state for host and CSE communication.<br>When this bit transitions from 0 to 1, hardware will clear the H_RDY and CSE_RDY bits. |
| 3 | 0x0<br>RW/V | **Host Ready (H_RDY):** This bit indicates that the host is ready to process messages. |
| 2 | 0x0<br>RW/V | **Host Interrupt Generate (H_IG):** Once message(s) are written into its CB, the host sets this bit to one for the HW to set the CSE_IS bit in the CSE_CSR and to generate an interrupt message to mIA. HW then clears this bit to 0. |
| 1 | 0x0<br>RW/1C/V | **Host Interrupt Status (H_IS):** HW sets this bit to 1 when CSE_IG bit is set to 1.<br>Host clears this bit to 0 by writing a 1 to this bit position.<br>H_IE has no effect on this bit. |
| 0 | 0x0<br>RW | **Host Interrupt Enable (H_IE):** Host sets this bit to 1 to enable the host interrupt (MSI, INTx, SMI or SCI) to be asserted when H_IS is set to 1. |

## 54.9.3 CSE Circular Buffer Read Window (HECI3_CSE_CB_RW)—Offset 8h

This register is for host to read from the CSE Circular Buffer (CSE_CB).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + 8h

**Default:** FFFFFFFFh

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:27, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xffffffff<br>RO | **CSE Circular Buffer Read Window Field (CSE_CB_RWF):** This bit field is for host to read from the CSE Circular Buffer.<br>This field is read only, writes have no effect.<br>Reads to this register will increment the CSE_CBRP as long as CSE_RDY is 1.<br>When CSE_RDY is 0, reads to this register have no effect, all 1s are returned, and CSE_CBRP is not incremented. |

## 54.9.4 CSE Control and Status Register Host Access (HECI3_CSE_CSR_HA)—Offset Ch

This register allows host software to read the CSE Control Status register (CSE_CSR).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [HECI3_MMIO_MBAR] + Ch

**Default:** 80000000h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:27, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x80<br>RO | **CSE Circular Buffer Depth Host Read Access (CSE_CBD_HRA):** Host read access to CSE_CBD.<br>**This field is hardwired to represent a depth of 128 entries.** |
| 23:16 | 0x0<br>RO/V | **CSE CB Write Pointer Host Read Access (CSE_CBWP_HRA):** Host read only access to CSE_CBWP. |
| 15:8 | 0x0<br>RO/V | **CSE CB Read Pointer Host Read Access (CSE_CBRP_HRA):** Host read only access to CSE_CBRP. |
| 7 | 0x0<br>RO/V | **NMI Status Host Read Access (NMI_STS_HRA):** Host read access to NMI_STS. |
| 6 | 0x0<br>RO | **Reserved (RSVD_6_6):** Reserved. |
| 5 | 0x0<br>RO/V | **Pointer Reset Host Read Access (PTR_RST_HRA):** Host read access to PTR_RST. |
| 4 | 0x0<br>RO/V | **CSE Reset Host Read Access (CSE_RST_HRA):** Host read access to CSE_RST. |
| 3 | 0x0<br>RO/V | **CSE Ready Host Read Access (CSE_RDY_HRA):** Host read access to CSE_RDY. |
| 2 | 0x0<br>RO/V | **CSE Interrupt Generate Host Read Access (CSE_IG_HRA):** Host read only access to CSE_IG. |
| 1 | 0x0<br>RO/V | **CSE Interrupt Status Host Read Access (CSE_IS_HRA):** Host read only access to CSE_IS. |
| 0 | 0x0<br>RO/V | **CSE Interrupt Enable Host Read Access (CSE_IE_HRA):** Host read only access to CSE_IE. |

## 54.9.5 D0i3 Control (HECI3_D0I3C)—Offset 800h

This register allows host to configure the power mode using D0i0/D0i3 support.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [HECI3_MMIO_MBAR] + 800h

**HECI3_MMIO_MBAR Type:** PCI Configuration Register (Size: 64 bits)
**HECI3_MMIO_MBAR Reference:** [B:0, D:27, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3 | 0x0<br>RO | **Restore Required (H_D0I3C_RR):** When this bit is set (by HW), SW must restore the state of the IP. This bit is cleared by SW writing a 1.<br>**In CSE this bit is never set so it does not need to be cleared by SW.** |
| 2 | 0x0<br>RW | **D0i3 (H_D0I3C_I3):** SW sets this bit to 1 in order to move the IP into the D0i3 state.<br>Clearing this bit will return the IP into the fully active D0 state (D0i0).<br>When this bit changes state, mIA may be interrupted (see H_PCI_CSR.D0I3C_IS register's description) |
| 1 | 0x0<br>RW | **Interrupt Request (H_D0I3C_IR):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this bit on each write to this register.<br>When this bit is set to 1, Command-in-Progress deassertion is captured in H_CSR.H_D0I3_IS.<br>If H_CSR.H_D0I3_IE is 1 as well, host interrupt will be initiated. |
| 0 | 0x0<br>RO/V | **Command-in-Progress (H_D0I3C_CIP):** HW sets this bit on a 0-)1 or 1-)0 transition of D0i3 bit (bit [2]).<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any of them.<br>While clear, all other bits in this register are valid and SW may write to them.<br>If Interrupt Request (bit [1]) was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to HW.<br>SW writes to this bit have no effect.<br>This bit is auto-cleared by HW one cycle after it is set. |

## 54.10 Registers in Memory Space—KT_HOST_MEMBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) KT_HOST_MEMBAR described in Section 54.5.6, "KT Memory BAR (KT_HOST_MEMBAR)—Offset 14h" on page 2513.

### 54.10.1 KT Receive Buffer Register (MEM_KTRBR)—Offset 0h

This register implements the standard Receiver Buffer register in the KT Block. This is read-only. DLAB(i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be read when the receiver data is ready. In FIFO mode, a read from this register will be translated into a read from the RBR FIFO. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Receiver Buffer Register (RBR):** Implements the Receiver Buffer register of the Serial Interface. |

### 54.10.2 KT Transmitter Holding Register (MEM_KTTHR)—Offset 0h

This register implements the standard Transmitter Holding register in the KT Block. This is write-only. DLAB(i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be written when the Host needs to transmit data. In FIFO mode, a write to this register will be translated into a write to the THR FIFO. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **Transmitter Holding Register (THR):** Implements the Transmit Data register of the Serial Interface. |

### 54.10.3 KT Divisor Latch LSB Register (MEM_KTDLLR)—Offset 0h

This register implements the standard DLL register in the KT Block. DLAB(i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. this register is supported only for SW compatibility and has no impact on HW performance. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 0h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch LSB (DLL):** Implements the DLL register of the Serial Interface. |

## 54.10.4 KT Interrupt Enable register (MEM_KTIER)—Offset 1h

This register implements the standard Interrupt Enable register in the KT Block. The bits enable specific events to interrupt the Host. See bit specific definition below. DLAB(i.e. KTLCR[7]) shall is 0 in order for this register to be accessed. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 1h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RW | **MSR (IER3):** When set, enables bits in Modem Status register to cause an interrupt to the Host |
| 2 | 0x0<br>RW | **LSR (IER2):** When set, enables bits in Receiver Line Status Register to cause an interrupt to the Host |
| 1 | 0x0<br>RW | **THR (IER1):** When set, enables an interrupt to the Host when the Transmitter Holding register is empty. |
| 0 | 0x0<br>RW | **DR (IER0):** When set, enables an interrupt to the Host when the Received Data is available OR Receiver FIFO timeout. |

## 54.10.5 KT Divisor Latch MSB Register (MEM_KTDLMR)—Offset 1h

This register implements the standard DLM register in the KT Block. DLAB(i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. This register is supported only for SW compatibility and has no impact on HW performance. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 1h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch MSB (DLM):** Implements the DLM register of the Serial Interface. |

## 54.10.6 KT Interrupt Identification Register (MEM_KTIIR)—Offset 2h

This register implements the standard Interrupt Identification register in the KT Block. This is read-only. This register is created to provide minimum software overhead during data character transfers. The register prioritizes KT interrupts into 4 levels and records them in the IIRSTS field of this register. The four levels of interrupt conditions in order of priority are: Receiver Line Status; Received Data Ready/Character Timeout Indication; Transmitter Holding Register Empty; MODEM Status. When the Host accesses the IIR, the HW freezes all interrupts and indicates the highest priority pending interrupt to the Host. While this Host access is occurring, the HW records new interrupts, but does not change its current indication until the access is complete. Table in the Host Interrupt Generation section shows the contents. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 2h

**Default:** 01h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **FIFO Enable (FIEN1):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 6 | 0x0<br>RO/V | **FIFO Enable (FIEN0):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0. |
| 3:1 | 0x0<br>RO/V | **IIR Status (IIRSTS):** These bits are asserted by the HW according to the source of the interrupt and the priority level.<br>Refer to the section on Host Interrupt Generation for a table of values. |
| 0 | 0x1<br>RO/V | **Interrupt Status (INTSTS):**<br>• 0: indicates a pending Host interrupt and IIRSTS may be used as a pointer to the appropriate interrupt service routine.<br>It reflects the logical-NOR of all UART interrupts.<br>An individual UART interrupt will be identified as pending when the individual event status is asserted AND its corresponding event interrupt enable is asserted as well.<br>For example, the THR Empty interrupt will be identified as pending when both LSR[THRE] bit is 1 AND IER[IER1] is 1.<br>• 1: indicates no pending Host interrupt. |

## 54.10.7 KT FIFO Control register (MEM_KTFCR)—Offset 2h

This register implements the standard FIFO Control register in the KT Block. It is used to configure features pertinent to the FIFO mode of operation. This is write-only. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 2h

**Default:** 00h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Receiver Trigger Level (RTL):** Trigger level in bytes for the Receiver FIFO.<br>Once the trigger level number of bytes is reached, an interrupt will be sent to the Host [7:6] RCVR FIFO Trigger Level (bytes)<br>• 00 -- 01<br>• 01 -- 04<br>• 10 -- 08<br>• 11 -- 14 |
| 5:4 | 0x0<br>WO | **Reserved (RSVD):** Writes to this field shall be dropped. |
| 3 | 0x0<br>WO | **RDY Mode (RDYM):** This bit has no effect on HW performance. |
| 2 | 0x0<br>WO | **XMT FIFO Clear (XFIC):** When the Host writes one to this bit the HW will clear the XMT FIFO.<br>This bit is self-cleared by HW. |
| 1 | 0x0<br>WO | **RCV FIFO Clear (RFIC):** When the Host writes one to this bit the HW will clear the RCV FIFO.<br>This bit is self-cleared by HW. |
| 0 | 0x0<br>WO | **FIFO Enable (FIE):** When set indicates that the KT interface is working in FIFO node.<br>When this bit value is changed the RCV and XMT FIFO are cleared by HW.<br>This bit must be a 1 when other FCR bits are written to or they will NOT be programmed. |

## 54.10.8 KT Line Control register (MEM_KTLCR)—Offset 3h

This register implements the standard Line Control register in the KT Block. It specifies the format of the asynchronous data communications exchange and sets the DLAB bit. Most bits in this register have no effect on HW and are only used by the FW. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 3h

**Default:** 03h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Divisor Latch Access Bit (DLAB):**<br>• 1: Allows access to the Divisor Latch Registers and reading of the FIFO Control Register.<br>• 0: Allows access to RBR, THR, IER and IIR registers. |
| 6 | 0x0<br>RW | **Break Control (BC):** This bit has no effect on HW. |
| 5:4 | 0x0<br>RW | **Parity Bit Mode (PBM):** This bit has no effect on HW. |
| 3 | 0x0<br>RW | **Parity Enable (PE):** This bit has no effect on HW. |
| 2 | 0x0<br>RW | **Stop Bit Select (SBS):** This bit has no effect on HW. |
| 1:0 | 0x3<br>RW | **Word Select Byte (WSB):** This bit has no effect on HW. |

## 54.10.9 KT Modem Control register (MEM_KTMCR)—Offset 4h

This register implements the standard Modem Control register in the KT Block. It controls the interface with the modem. Because the FW emulates the modem, the Host communicates with the FW via this register. This register has impact on the HW in the loopback mode. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 4h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4 | 0x0<br>RW | **Loop Back Mode (LBM):** When set by Host indicates that the serial port is in loop Back mode. This means that the data that is transmitted by the host should be received.<br>Helps in debug of the interface |
| 3 | 0x0<br>RW | **Output 2 (OUT2):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit (7). |
| 2 | 0x0<br>RW | **Output 1 (OUT1):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit(6). |
| 1 | 0x0<br>RW | **Request to Send Out (RTSO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value of this bit is written by HW to Modem Status Register bit(4). |
| 0 | 0x0<br>RW | **Data Terminal Ready Out (DRTO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value in this bit is written by HW to Modem Status Register Bit(5). |

## 54.10.10 KT Line Status register (MEM_KTLSR)—Offset 5h

This register implements the standard Line Status register in the KT Block. It provides status information of the data transfer to the Host. Error indication etc are provided by the HW/FW to the host via this register. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [KT_HOST_MEMBAR] + 5h

**Default:** 60h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/C/V | **RCVR FIFO Error (RXFER):** This bit is cleared in non FIFO mode.<br>This bit is connected to UART LSR[BI] bit in FIFO mode. |
| 6 | 0x1<br>RO/V | **Transmitter Empty (TEMT):**<br>In non-FIFO mode:<br>This bit will be set to 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, OR cleared to 0 whenever either THR or TSR contains a data byte.<br><br>In FIFO mode:<br>This bit will be set to 1 whenever XMIT FIFO and TSR are both empty, OR cleared to 0 whenever either XMIT FIFO or TSR contains a data byte.<br><br>Implementation note: This bit is connected by HW to THRE bit |
| 5 | 0x1<br>RO/V | **Transmitter Holding Register Empty (THRE):**<br>The bit will always be set whenever Host changes UART FCR[FIE] bit.<br>This bit is controlled differently in FIFO vs non-FIFO modes:<br>Non-FIFO mode:<br>This bit will be cleared by HW when Host writes to THR, and set by HW when FW reads from THR.<br><br>FIFO mode:<br>This bit will be set by HW when XMIT FIFO is empty , and cleared by HW when XMIT FIFO is NOT empty. |
| 4 | 0x0<br>RO/C/V | **Break Interrupt (BI):** This bit will be cleared by HW when the UART LSR register is read by Host.<br>This bit will be set by HW in two cases:<br>In FIFO mode, when FW sets the KTRIVR[SBI] bit.<br>In non-FIFO mode, when FW sets the KTRBR[BIA] bit. |
| 3 | 0x0<br>RO | **Framing Error (FE):** This bit is NOT implemented and hardwired to 0. |
| 2 | 0x0<br>RO | **Parity Error (PE):** This bit is NOT implemented and hardwired to 0. |
| 1 | 0x0<br>RO/C/V | **Overrun Error (OE):** HW will clear this bit when Host reads from KTLSR register and this is source of interrupt.<br>HW will set this bit when overrun error is detected.<br>In non-FIFO mode, the character that triggers the overrun error shall overwrite the character in the RBR register.<br>In FIFO mode, the character that triggers the overrun error shall be dropped without corrupting the RCVR FIFO. |
| 0 | 0x0<br>RO/V | **Data Ready (DR):**<br>Non-FIFO Mode:<br>This bit will be set by HW when FW writes to RBR and cleared by HW when Host reads from RBR.<br>FIFO Mode:<br>The bit will be set by HW when RCVR FIFO is not empty and cleared by HW when RCVR FIFO is empty. |

## 54.10.11 KT Modem Status register (MEM_KTMSR)—Offset 6h

This register implements the standard Modem Status register in the KT Block. It provides the current state of the control lines from the modem. The modem functionality is emulated by the FW. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 6h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Data Carrier Detect (DCD):** In Loop Back mode this bit is connected by HW to the value of MCR bit 3 |
| 6 | 0x0<br>RO | **Ring Indicator (RI):** In Loop Back mode this bit is connected by HW to the value of MCR bit 2. |
| 5 | 0x0<br>RO | **Data Set Ready (DSR):** In Loop Back mode this bit is connected by HW to the value of MCR bit 0. |
| 4 | 0x0<br>RO | **Clear To Send (CTS):** In Loop Back mode this bit is connected by HW to the value of MCR bit 1. |
| 3 | 0x0<br>RO/C/V | **Delta Data Carrier Detect (DDCD):** This bit is set when bit 7 is changed.<br>This bit is cleared by HW when the MSR register is being read by the HOST driver. |
| 2 | 0x0<br>RO/C/V | **Trailing Edge of Read Detector (TERI):** This bit is set when bit 6 is changed from 1 to 0.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 1 | 0x0<br>RO/C/V | **Delta Data Set Ready (DDSR):** This bit is set when bit 5 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 0 | 0x0<br>RO/C/V | **Delta Clear To Send (DCTS):** This bit is set when bit 4 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |

## 54.10.12 KT Scratch register (MEM_KTSCR)—Offset 7h

This register implements the standard Scratch register in the KT Block. It allows Host SW programmers to hold temporary data. It has NO effect on HW. Reset condition: Host reset. Non-sticky.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_MEMBAR] + 7h

**KT_HOST_MEMBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_MEMBAR Reference:** [B:0, D:27, F:3] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Scratch Register Data (SCRD):** Reserved. |

## 54.11 Registers in Memory Space—Fixed Address

These registers are accessed in Host Memory Space through fixed memory addresses.

### 54.11.1 FTPM Locality State (HOST_LOC_0_LOC_STATE_HROA)—Offset FED40000h

Reads by host SW from this register will reflect the value of FTPM_LOC_STATE register in mIA space. This register is aliased to the following addresses in the host address space: FED4_x000, x=0,1,2,3,4. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40000h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO/V | **Reserved (RSVD_31_8):** Opaque to HW. Defined by FW. |
| 7 | 0x0<br>RO/V | **Register Valid Status (FTPM_REG_VALID_STS):** Opaque to HW.<br>This bit is used by FW to signal to host SW when this register is valid. |
| 6:5 | 0x0<br>RO/V | **Reserved (RSVD_6_5):** Opaque to HW. Defined by FW. |
| 4:2 | 0x0<br>RO/V | **Active Locality (FTPM_ACTV_LOC):** The encodings are:<br>• 000 - Locality 0,<br>• 001 - Locality 1,<br>• 010 - Locality 2,<br>• 011 - Locality 3,<br>• 100 - Locality 4,<br>• 101 and 111 - reserved.<br>FW sets these bits to mux in the active locality.<br>Once set, the FTPM HCI will pass cycles from the active host locality to the FTPM shared SRAM.<br>Writes from in-active localities will be dropped and reads will return 0's. |
| 1 | 0x0<br>RO/V | **Locality Assigned (FTPM_LOC_ASSIGNED):** A value of 0 indicates to host SW that no locality is assigned, a value of 1 indicates a locality has been assigned. |
| 0 | 0x0<br>RO/V | **Establishment (FTPM_ESTABLISHMENT):**<br>Opaque to HW.<br>This bit is set to 0 by FW when HASH_START occurs and set to 1 when the RESET_ESTABLISHMENT bit is set. |

### 54.11.2 FTPM Locality Reserved (HOST_LOC_0_LOC_RSVD)—Offset FED40004h

This register was added in order to support QW reads from Locality State register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40004h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Reserved (FTPM_LOC_RSVD_BITS):** Reserved. |

### 54.11.3 FTPM Locality n Control Register (HOST_LOC_0_LOC_CTRL)—Offset FED40008h

Writes by host SW to this register will update the value of the corresponding LOC_n_CTRL_CSEROA register in mIA space. Writes by host SW to this register may be configured by CSE FW to trigger an MSI/PME to mIA. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40008h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>WO | **FTPM Locality n Control Bits (FTPM_LOC_CTRL_BITS):** Reserved. |

### 54.11.4 FTPM Locality n Status Register Host Read Only Access. (HOST_LOC_0_LOC_STS_HROA)—Offset FED4000Ch

Reads by host SW from this register will reflect the value of the corresponding LOC_n_STS register in mIA space. This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4000Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Locality n Status Bits (FTPM_LOC_STS_BITS):** Reserved. |

### 54.11.5 FTPM Locality Config Area Reserved Read Write[0-7] (HOST_LOC_0_RSVD_RW1[0-7])—Offset FED40010h, Count 8, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. This DW is aliased to the same offset for each locality, and even inactive localities can access it.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-7]:** [0x0] + FED40010h + [0-7]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Locality Config Area Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

### 54.11.6 FTPM Interface ID Host Read Only Access 0 (HOST_LOC_0_INTF_ID_0)—Offset FED40030h

This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40030h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Interface ID Bits 0 (FTPM_INTF_ID_BITS_0):** These are the 32 LSbits used by FW to inform host SW as to the interface type of the TPM.<br>The exact usage is defined by TCG. |

### 54.11.7 FTPM Interface ID Host Read Only Access 1 (HOST_LOC_0_INTF_ID_1)—Offset FED40034h

This register is opaque to HW. This register is implemented in flip-flops.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40034h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **FTPM Interface ID Bits 1 (FTPM_INTF_ID_BITS_1):** These are the 32 MSbits used by FW to inform host SW as to the interface type of the TPM.<br>The exact usage is defined by TCG. |

### 54.11.8 FTPM Locality Config Area Reserved Read Write[0-1] (HOST_LOC_0_RSVD_RW2[0-1])—Offset FED40038h, Count 2, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. This DW is aliased to the same offset for each locality, and even inactive localities can access it.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-1]:** [0x0] + FED40038h + [0-1]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Locality Config Area Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

## 54.11.9 Control Area Request (HOST_LOC_0_CA_REQUEST)—Offset FED40040h

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40040h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_REQUEST:** Host software sets this field to a CSxE firmware-defined value to indicate a state change request to the fTPM.<br>Any host write to this register shall result in an interrupt to the CSxE firmware, or wakeup the CSxE subsystem when it is in power-gated state. |

## 54.11.10 Control Area Status (HOST_LOC_0_CA_STATUS)—Offset FED40044h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40044h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_STATUS:** Reserved. |

## 54.11.11 Control Area Cancel (HOST_LOC_0_CA_CANCEL)—Offset FED40048h

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40048h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CANCEL:** Host software sets this field to a CSxE firmware-defined value to indicate that a command should be canceled.<br>Any host write to this register shall result in an interrupt to the CSxE firmware, or wakeup the CSxE subsystem when it is in power-gated state. |

## 54.11.12 Control Area Start (HOST_LOC_0_CA_START)—Offset FED4004Ch

This register is implemented in flip-flops. This register is implemented in retention flops to maintain state through PG.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4004Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Control Area Start (HOST_CA_START):** Host software sets this field to a CSxE firmware defined value to indicate that a command is available for processing.<br>The reset value indicates that a command is NOT available for processing.<br>Any host write to this register shall result in an interrupt to the CSE firmware or wakeup the CSE subsystem when it is in power-gated state. |

## 54.11.13 Control Area Interrupt Reserved Part 1 (HOST_LOC_0_CA_INT_RSVD1)—Offset FED40050h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40050h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RSVD_31_0:** Reserved. |

## 54.11.14 Control Area Interrupt Reserved Part 2 (HOST_LOC_0_CA_INT_RSVD2)—Offset FED40054h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40054h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RSVD_31_0:** Reserved. |

## 54.11.15 Control Area Command Size (HOST_LOC_0_CA_CMD_SZ)—Offset FED40058h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40058h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD_SZ:** Reserved. |

## 54.11.16 Control Area Command Part 1 (HOST_LOC_0_CA_CMD1)—Offset FED4005Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4005Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD1:** Reserved. |

## 54.11.17 Control Area Command Part 2 (HOST_LOC_0_CA_CMD2)—Offset FED40060h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40060h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_CMD2:** Reserved. |

## 54.11.18 Control Area Response Size (HOST_LOC_0_CA_RSP_SZ)—Offset FED40064h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40064h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP_SZ:** Reserved. |

## 54.11.19 Control Area Response Part 1 (HOST_LOC_0_CA_RSP1)—Offset FED40068h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED40068h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP1:** Reserved. |

## 54.11.20 Control Area Response Part 2 (HOST_LOC_0_CA_RSP2)—Offset FED4006Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [0x0] + FED4006Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **HOST_CA_RSP2:** Reserved. |

## 54.11.21 FTPM Reserved Read Write[0-3] (HOST_LOC_0_RSVD_RW3[0-3])—Offset FED40070h, Count 4, Stride 4h

This register represents a DW of RW memory in FTPM EBB which has no defined functionality. The access to this DW is allowed only to an active locality.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-3]:** [0x0] + FED40070h + [0-3]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Reserved Read Write DW (RSVD_RW_31_0):** Reserved. |

## 54.11.22 FTPM Command and Response Buffer[0-991] (HOST_LOC_0_FTPM_CRB[0-991])—Offset FED40080h, Count 992, Stride 4h

This register represents a DW of RW memory in FTPM CRB (EBB)

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset[0-991]:** [0x0] + FED40080h + [0-991]*4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **FTPM Command and Response Buffer DW (HOST_FTPM_CRB_DW):** Reserved. |

## 54.12 Registers in I/O Space—KT_HOST_IOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) KT_HOST_IOBAR described in Section 54.5.5, "KT IO BAR (KT_HOST_IOBAR)—Offset 10h" on page 2512.

### 54.12.1 KT Receive Buffer Register (IO_KTRBR)—Offset 0h

This register implements the standard Receiver Buffer register in the KT Block. This is read-only. DLAB(i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be read when the receiver data is ready. In FIFO mode, a read from this register will be translated into a read from the RBR FIFO. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Receiver Buffer Register (RBR):** Implements the Receiver Buffer register of the Serial Interface. |

### 54.12.2 KT Transmitter Holding Register (IO_KTTHR)—Offset 0h

This register implements the standard Transmitter Holding register in the KT Block. This is write-only. DLAB(i.e. KTLCR[7]) shall be 0. In non-FIFO mode, this register will be written when the Host needs to transmit data. In FIFO mode, a write to this register will be translated into a write to the THR FIFO. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **Transmitter Holding Register (THR):** Implements the Transmit Data register of the Serial Interface. |

### 54.12.3 KT Divisor Latch LSB Register (IO_KTDLLR)—Offset 0h

This register implements the standard DLL register in the KT Block. DLAB(i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. this register is supported only for SW compatibility and has no impact on HW performance. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 0h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch LSB (DLL):** Implements the DLL register of the Serial Interface. |

### 54.12.4 KT Interrupt Enable register (IO_KTIER)—Offset 1h

This register implements the standard Interrupt Enable register in the KT Block. The bits enable specific events to interrupt the Host. See bit specific definition below. DLAB(i.e. KTLCR[7]) shall is 0 in order for this register to be accessed. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 1h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x0<br>RW | **MSR (IER3):** When set, enables bits in Modem Status register to cause an interrupt to the Host. |
| 2 | 0x0<br>RW | **LSR (IER2):** When set, enables bits in Receiver Line Status Register to cause an interrupt to the Host. |
| 1 | 0x0<br>RW | **THR (IER1):** When set, enables an interrupt to the Host when the Transmitter Holding register is empty. |
| 0 | 0x0<br>RW | **DR (IER0):** When set, enables an interrupt to the Host when the Received Data is available OR Receiver FIFO timeout. |

## 54.12.5 KT Divisor Latch MSB Register (IO_KTDLMR)—Offset 1h

This register implements the standard DLM register in the KT Block. DLAB(i.e. KTLCR[7]) shall be 1 in order for this register to be accessed. This register is supported only for SW compatibility and has no impact on HW performance. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 1h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Divisor Latch MSB (DLM):** Implements the DLM register of the Serial Interface. |

## 54.12.6 KT Interrupt Identification Register (IO_KTIIR)—Offset 2h

This register implements the standard Interrupt Identification register in the KT Block. This is read-only. This register is created to provide minimum software overhead during data character transfers. The register prioritizes KT interrupts into 4 levels and records them in the IIRSTS field of this register. The four levels of interrupt conditions in order of priority are: Receiver Line Status; Received Data Ready/Character Timeout Indication; Transmitter Holding Register Empty; MODEM Status. When the Host accesses the IIR, the HW freezes all interrupts and indicates the highest priority pending interrupt to the Host. While this Host access is occurring, the HW records new interrupts, but does not change its current indication until the access is complete. Table in the Host Interrupt Generation section shows the contents. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 2h

**Default:** 01h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **FIFO Enable (FIEN1):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 6 | 0x0<br>RO/V | **FIFO Enable (FIEN0):** This field is connected by HW to the FIFO Enable field of the FCR register. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD):** Hardwired to 0. |
| 3:1 | 0x0<br>RO/V | **IIR Status (IIRSTS):** These bits are asserted by the HW according to the source of the interrupt and the priority level.<br>Refer to the section on Host Interrupt Generation for a table of values. |
| 0 | 0x1<br>RO/V | **Interrupt Status (INTSTS):**<br>• 0: indicates a pending Host interrupt and IIRSTS may be used as a pointer to the appropriate interrupt service routine.<br>It reflects the logical-NOR of all UART interrupts.<br>An individual UART interrupt will be identified as pending when the individual event status is asserted AND its corresponding event interrupt enable is asserted as well.<br>For example, the THR Empty interrupt will be identified as pending when both LSR[THRE] bit is 1 AND IER[IER1] is 1.<br>• 1: indicates no pending Host interrupt. |

## 54.12.7 KT FIFO Control register (IO_KTFCR)—Offset 2h

This register implements the standard FIFO Control register in the KT Block. It is used to configure features pertinent to the FIFO mode of operation. This is write-only. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 2h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Receiver Trigger Level (RTL):** Trigger level in bytes for the Receiver FIFO.<br>Once the trigger level number of bytes is reached, an interrupt will be sent to the Host [7:6] RCVR FIFO Trigger Level (bytes)<br>• 00 -- 01<br>• 01 -- 04<br>• 10 -- 08<br>• 11 -- 14 |
| 5:4 | 0x0<br>WO | **Reserved (RSVD):** Writes to this field shall be dropped. |
| 3 | 0x0<br>WO | **RDY Mode (RDYM):** This bit has no effect on HW performance. |
| 2 | 0x0<br>WO | **XMT FIFO Clear (XFIC):** When the Host writes one to this bit the HW will clear the XMT FIFO.<br>This bit is self-cleared by HW. |
| 1 | 0x0<br>WO | **RCV FIFO Clear (RFIC):** When the Host writes one to this bit the HW will clear the RCV FIFO.<br>This bit is self-cleared by HW. |
| 0 | 0x0<br>WO | **FIFO Enable (FIE):** When set indicates that the KT interface is working in FIFO node.<br>When this bit value is changed the RCV and XMT FIFO are cleared by HW.<br>This bit must be a 1 when other FCR bits are written to or they will NOT be programmed. |

## 54.12.8 KT Line Control register (IO_KTLCR)—Offset 3h

This register implements the standard Line Control register in the KT Block. It specifies the format of the asynchronous data communications exchange and sets the DLAB bit. Most bits in this register have no effect on HW and are only used by the FW. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 3h

**Default:** 03h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Divisor Latch Access Bit (DLAB):**<br>• 1: Allows access to the Divisor Latch Registers and reading of the FIFO Control Register.<br>• 0: Allows access to RBR, THR, IER and IIR registers. |
| 6 | 0x0<br>RW | **Break Control (BC):** This bit has no effect on HW. |
| 5:4 | 0x0<br>RW | **Parity Bit Mode (PBM):** This bit has no effect on HW. |
| 3 | 0x0<br>RW | **Parity Enable (PE):** This bit has no effect on HW. |
| 2 | 0x0<br>RW | **Stop Bit Select (SBS):** This bit has no effect on HW. |
| 1:0 | 0x3<br>RW | **Word Select Byte (WSB):** This bit has no effect on HW. |

## 54.12.9 KT Modem Control register (IO_KTMCR)—Offset 4h

This register implements the standard Modem Control register in the KT Block. It controls the interface with the modem. Because the FW emulates the modem, the Host communicates with the FW via this register. This register has impact on the HW in the loopback mode. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 4h

**Default:** 00h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4 | 0x0<br>RW | **Loop Back Mode (LBM):** When set by Host indicates that the serial port is in loop Back mode.<br>This means that the data that is transmitted by the host should be received.<br>Helps in debug of the interface. |
| 3 | 0x0<br>RW | **Output 2 (OUT2):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit (7). |
| 2 | 0x0<br>RW | **Output 1 (OUT1):** This bit has no effect on HW in normal mode.<br>In loop back mode the value of this bit is written by HW to Modem Status Register bit(6). |
| 1 | 0x0<br>RW | **Request to Send Out (RTSO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value of this bit is written by HW to Modem Status Register bit(4). |
| 0 | 0x0<br>RW | **Data Terminal Ready Out (DRTO):** This bit has no effect on HW in normal mode.<br>In loopback mode, the value in this bit is written by HW to Modem Status Register Bit(5). |

## 54.12.10 KT Line Status register (IO_KTLSR)—Offset 5h

This register implements the standard Line Status register in the KT Block. It provides status information of the data transfer to the Host. Error indication etc are provided by the HW/FW to the host via this register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [KT_HOST_IOBAR] + 5h

**Default:** 60h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/C/V | **RCVR FIFO Error (RXFER):** This bit is cleared in non FIFO mode.<br>This bit is connected to UART LSR[BI] bit in FIFO mode. |
| 6 | 0x1<br>RO/C/V | **Transmitter Empty (TEMT):**<br>In non-FIFO mode:<br>This bit will be set to 1 whenever the Transmitter Holding Register (THR) and the Transmitter Shift Register (TSR) are both empty, OR cleared to 0 whenever either THR or TSR contains a data byte.<br><br>In FIFO mode:<br>This bit will be set to 1 whenever XMIT FIFO and TSR are both empty, OR cleared to 0 whenever either XMIT FIFO or TSR contains a data byte.<br><br>Implementation note: This bit is connected by HW to THRE bit |
| 5 | 0x1<br>RO/C/V | **Transmitter Holding Register Empty (THRE):** The bit will always be set whenever Host changes UART FCR[FIE] bit.<br>This bit is controlled differently in FIFO vs non-FIFO modes:<br>Non-FIFO mode:<br>This bit will be cleared by HW when Host writes to THR, and set by HW when FW reads from THR.<br><br>FIFO mode:<br>This bit will be set by HW when XMIT FIFO is empty , and cleared by HW when XMIT FIFO is NOT empty. |
| 4 | 0x0<br>RO/C/V | **Break Interrupt (BI):** This bit will be cleared by HW when the UART LSR register is read by Host.<br>This bit will be set by HW in two cases:<br>In FIFO mode, when FW sets the KTRIVR[SBI] bit.<br>In non-FIFO mode, when FW sets the KTRBR[BIA] bit. |
| 3 | 0x0<br>RO | **Framing Error (FE):** This bit is NOT implemented and hardwired to 0. |
| 2 | 0x0<br>RO | **Parity Error (PE):** This bit is NOT implemented and hardwired to 0. |
| 1 | 0x0<br>RO/C/V | **Overrun Error (OE):** HW will clear this bit when Host reads from KTLSR register and this is source of interrupt.<br>HW will set this bit when overrun error is detected.<br>In non-FIFO mode, the character that triggers the overrun error shall overwrite the character in the RBR register.<br>In FIFO mode, the character that triggers the overrun error shall be dropped without corrupting the RCVR FIFO. |
| 0 | 0x0<br>RO/V | **Data Ready (DR):**<br>Non-FIFO Mode:<br>This bit will be set by HW when FW writes to RBR and cleared by HW when Host reads from RBR.<br><br>FIFO Mode:<br>The bit will be set by HW when RCVR FIFO is not empty and cleared by HW when RCVR FIFO is empty. |

## 54.12.11 KT Modem Status register (IO_KTMSR)—Offset 6h

This register implements the standard Modem Status register in the KT Block. It provides the current state of the control lines from the modem. The modem functionality is emulated by the FW. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 6h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Data Carrier Detect (DCD):** In Loop Back mode this bit is connected by HW to the value of MCR bit 3. |
| 6 | 0x0<br>RO | **Ring Indicator (RI):** In Loop Back mode this bit is connected by HW to the value of MCR bit 2. |
| 5 | 0x0<br>RO | **Data Set Ready (DSR):** In Loop Back mode this bit is connected by HW to the value of MCR bit 0. |
| 4 | 0x0<br>RO | **Clear To Send (CTS):** In Loop Back mode this bit is connected by HW to the value of MCR bit 1. |
| 3 | 0x0<br>RO/C/V | **Delta Data Carrier Detect (DDCD):** This bit is set when bit 7 is changed.<br>This bit is cleared by HW when the MSR register is being read by the HOST driver. |
| 2 | 0x0<br>RO/C/V | **Trailing Edge of Read Detector (TERI):** This bit is set when bit 6 is changed from 1 to 0.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 1 | 0x0<br>RO/C/V | **Delta Data Set Ready (DDSR):** This bit is set when bit 5 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |
| 0 | 0x0<br>RO/C/V | **Delta Clear To Send (DCTS):** This bit is set when bit 4 is changed.<br>This bit is cleared by HW when the MSR register is being read by the Host driver. |

## 54.12.12 KT Scratch register (IO_KTSCR)—Offset 7h

This register implements the standard Scratch register in the KT Block. It allows Host SW programmers to hold temporary data. It has NO effect on HW. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [KT_HOST_IOBAR] + 7h

**KT_HOST_IOBAR Type:** PCI Configuration Register (Size: 32 bits)
**KT_HOST_IOBAR Reference:** [B:0, D:27, F:3] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Scratch Register Data (SCRD):** Reserved. |

## 54.13 Registers in I/O Space—IDE_HOST_PCMDIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_PCMDIOBAR described in Section 54.6.5, "IDE Primary Command Block IO BAR (IDE_HOST_PCMDIOBAR)—Offset 10h" on page 2525.

### 54.13.1 IDE Data Register (IDEDATA)—Offset 0h

This register serves as the data interface in IDE PIO modes. Unit of access is in WORD. Host writes to this register are converted into writes to the PIO buffer in CSxE memory at monotonically increasing WORD address. Host writes are dropped under any one of the following conditions: (1) PIO modes are disabled. (Cfg. IDEPBPR[EN] defined in CSxE memory space) (2) PIO buffer is full. Host reads from this register are converted into reads from the PIO buffer in CSxE memory at monotonically increasing WORD address. Host reads are dropped under any one of the following conditions: (1) PIO modes are disabled. (Cfg. IDEPBPR[EN] defined in CSxE memory space) (2) PIO buffer is empty. Reset condition: no connection to any reset signal.

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 0h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/V | **IDE Data Register (IDEDR):** Data interface in IDE PIO modes.<br>Host reads/writes from/to this register are converted into reads/writes from/to the PIO buffer in CSxE memory. |

### 54.13.2 IDE Features Register (IDEFR)—Offset 1h

This register implements the standard Features register in the Primary IDE Command Block. This is write-only. BE[0] shall be deasserted. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Features Data (IDEFD):** The IDE Features register is command specific and is used to enable and disable interface features. |

## 54.13.3 IDE Error Register DEV0 (IDEERD0)—Offset 1h

This register implements the standard Device 0 Error register in the Primary IDE Command Block. This is read-only. BE[0] shall be deasserted. DEV = 0 (i.e. IDEDHIR[4] = 0). Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Error Data (IDEED):** Upon command completion of any command except EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET, the content of this register is valid and contains error status when the ERR bit is set to 1 in the Status register (i.e. IDESR0[ERR] = 1).<br>Following a power-on, a hardware or software reset, or command completion of an EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET command, this register contains a diagnostic code.<br>Bit 2 ABRT (command aborted) is set to 1 to indicate the requested command has been command aborted because the command code or a command parameter is invalid, the command is not supported, a pre-requisite for the command has not been met, or some other error has occurred. |

## 54.13.4 IDE Error Register DEV1 (IDEERD1)—Offset 1h

This register implements the standard Device 1 Error register in the Primary IDE Command Block. This is read-only. BE[0] shall be deasserted. DEV = 1 (i.e. IDEDHIR[4] = 1). Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 1h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Error Data (IDEED):** Upon command completion of any command except EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET, the content of this register is valid and contains error status when the ERR bit is set to 1 in the Status register (i.e. IDESR0[ERR] = 1).<br>Following a power-on, a hardware or software reset, or command completion of an EXECUTE DEVICE DIAGNOSTIC or DEVICE RESET command, this register contains a diagnostic code.<br>Bit 2 ABRT (command aborted) is set to 1 to indicate the requested command has been command aborted because the command code or a command parameter is invalid, the command is not supported, a pre-requisite for the command has not been met, or some other error has occurred. |

## 54.13.5 IDE Sector Count In Register (IDESCIR)—Offset 2h

This register implements the standard Sector Count In register in the Primary IDE Command Block. All 3 registers (IDESCIR, IDESCOR0, IDESCOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is meaningful for an ATA command while unused for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Sector Count Data (IDESCD):** The Sector Count register specifies the sector count of the data for which an ATA transfer has been made. |

## 54.13.6 IDE Sector Count Out Register DEV0 (IDESCOR0)—Offset 2h

This register implements the standard Device 0 Sector Count Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA/ATAPI command. This register is called IDE Interrupt Reason Register DEV0 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Count Out DEV0 (ISCOD0):** Sector Count Out register for Device 0. |

## 54.13.7 IDE Sector Count Out Register DEV1 (IDESCOR1)—Offset 2h

This register implements the standard Device 1 Sector Count Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA/ATAPI command. This register is called IDE Interrupt Reason Register DEV1 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 2h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Count Out DEV1 (ISCOD1):** Sector Count Out register for Device 1. |

## 54.13.8 IDE Sector Number In Register (IDESNIR)—Offset 3h

This register implements the standard Sector Number In register in the Primary IDE Command Block in CHS addressing mode. All 3 registers (IDESNIR, IDESNOR0, IDESNOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA Low In Register in LBA addressing mode. This register is reserved for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Sector Number Data (IDESND):** The Sector Number register is set by the host to specify the starting sector number associated with the next ATA command to be executed.<br>Following a qualified ATA command sequence the device will set the register value to the last sector read or written as a result of the previous ATA command. |

## 54.13.9 IDE Sector Number Out Register DEV0 (IDESNOR0)—Offset 3h

This register implements the standard Device 0 Sector Number Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Low Out Register DEV0 in LBA addressing mode. This register is reserved for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Number Out DEV0 (IDESNO0):** Sector Number Out register for Device 0. |

## 54.13.10 IDE Sector Number Out Register DEV1 (IDESNOR1)—Offset 3h

This register implements the standard Device 1 Sector Number Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Low Out Register DEV1 in LBA addressing mode. This register is reserved for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 3h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Sector Number Out DEV1 (IDESNO1):** Sector Number Out register for Device 1. |

## 54.13.11 IDE Cylinder Low In Register (IDECLIR)—Offset 4h

This register implements the standard Cylinder Low In register in the Primary IDE Command Block. All 3 registers (IDECLIR, IDCLOR0, IDCLOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA Mid In Register in LBA addressing mode. This register is called IDE Byte Count Low Limit Register for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Cylinder Low Data. (IDECLD):** The Cylinder Low register is set by the host to specify the cylinder number low byte.<br>Following an ATA command, the contents of this register is written by the device, identifying the cylinder number low byte. |

## 54.13.12 IDE Cylinder Low Out Register DEV0 (IDCLOR0)—Offset 4h

This register implements the standard Device 0 Cylinder Low Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Mid Out Register DEV0 in LBA addressing mode. This register is called IDE Byte Count Low Register DEV0 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder Low Out DEV0. (IDECLO0):** Cylinder Low Out Register for Device 0. |

## 54.13.13 IDE Cylinder Low Out Register DEV1 (IDCLOR1)—Offset 4h

This register implements the standard Device 1 Cylinder Low Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA Mid Out Register DEV1 in LBA addressing mode. This register is called IDE Byte Count Low Register DEV1 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 4h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder Low Out DEV1. (IDECLO1):** Cylinder Low Out Register for Device 1. |

## 54.13.14 IDE Cylinder High In Register (IDECHIR)—Offset 5h

This register implements the standard Cylinder High In register in the Primary IDE Command Block. All 3 registers (IDECHIR, IDCHOR0, IDCHOR1) will be updated with the same write data when this register is write-accessed. This is write-only. This register is called IDE LBA High In Register in LBA addressing mode. This register is called IDE Byte Count High Limit Register for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Cylinder High Data (IDECHD):** The Cylinder High register is set by the host to specify the cylinder number high byte.<br>Following an ATA command, the contents of this register is written by the device, identifying the cylinder number high byte. |

## 54.13.15 IDE Cylinder High Out Register DEV0 (IDCHOR0)—Offset 5h

This register implements the standard Device 0 Cylinder High Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA High Out Register DEV0 in LBA addressing mode. This register is called IDE Byte Count High Register DEV0 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder High Out DEV0 (IDECHO0):** Cylinder High Out register for Device 0. |

## 54.13.16 IDE Cylinder High Out Register DEV1 (IDCHOR1)—Offset 5h

This register implements the standard Device 1 Cylinder High Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE LBA High Out Register DEV1 in LBA addressing mode. This register is called IDE Byte Count High Register DEV1 for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 5h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Cylinder High Out DEV1 (IDECHO1):** Cylinder High Out register for Device 1. |

## 54.13.17 IDE Drive/Head In Register (IDEDHIR)—Offset 6h

This register implements the standard Drive/Head In register in the Primary IDE Command Block. All 3 registers (IDEDHIR, IDDHOR0, IDDHOR1) will be updated with the same write data when this register is write-accessed. This is write-only. Bit 4 of this register is the DEV (master/slave) bit. This bit is cleared by hardware on IDE software reset (IDDCR[SRST] = 1) in addition to Host reset. This register is called IDE Device In Register in LBA addressing mode or for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Drive/Head Data (IDEDHD):** The Drive/Head Register is used by the host and the device to select the type of addressing (CHS or LBA), the drive letter, and either bits 3 through 0 of the head number in CHS mode or in LBA mode, bits D3:D0 reflect the logical block number bits LBA27:LBA24. |

## 54.13.18 IDE Drive/Head Out Register DEV0 (IDDHOR0)—Offset 6h

This register implements the standard Device 0 Drive/Head Out register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE Device Out Register DEV0 in LBA addressing mode or for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Drive/Head Out DEV0 (IDEDHO0):** Drive/Head Out register of Device 0. |

## 54.13.19 IDE Drive/Head Out Register DEV1 (IDDHOR1)—Offset 6h

This register implements the standard Device 1 Drive/Head Out register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA command. This register is called IDE Device Out Register DEV1 in LBA addressing mode or for an ATAPI command. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 6h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Drive/Head Out DEV1 (IDEDHO1):** Drive/Head Out register of Device 1. |

## 54.13.20 IDE Command Register (IDECR)—Offset 7h

This register implements the standard Command register in the Primary IDE Command Block. This is write-only. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**Default:** 00h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>WO | **IDE Command Data (IDECD):** The Command register specifies the ATA/ATAPI Command code being issued to the drive by the host. Command processing begins when this register is written.<br>The content of the Command Block registers becomes parameters of the command when this register is written.<br>Writing this register clears any pending interrupt condition.<br>Please refer to the description of the INTM and INTS bits in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. |

## 54.13.21 IDE Status Register DEV0 (IDESR0)—Offset 7h

This register implements the standard Device 0 Status register in the Primary IDE Command Block. This is read-only. DEV = 0 (i.e. IDEDHIR[4] = 0). CSxE FW writes to this register upon completion of an ATA/ATAPI command. Additionally upon completion of the last data phase of a PIO transfer, HW will load this register with value of the SRAFP field in the IDESRAP register in the CSxE memory space. Host read of this register clears Device 0's interrupt. Please refer to the description of the INTM bit in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. The bit description is for ATA mode. Different name in ATAPI mode, if applicable, is cited as well. Reset condition: CSxE reset with the BSY bit as an exception. Please refer to description of the BSY bit below for a list of BSY bit reset conditions. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**Default:** 80h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Busy (BSY):** This bit is set by HW when the IDECR is being written and DEV=0, OR when IDDCR[SRST] bit is asserted by Host, OR host system reset, OR D3 to D0 transition of the IDE function.<br>This bit is cleared by FW write of 0.<br>Please refer to general register description above for special handling for a PIO transfer.<br>When this bit is 1, any other bit in this register is meaningless. |
| 6 | 0x0<br>RO/V | **Drive Ready (DRDY):** When set, indicates drive is ready for command.<br>Update of this bit is completely FW-assisted.<br>This bit is also called Device Ready.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 5 | 0x0<br>RO/V | **Drive Fault (DF):** When set, indicates Error on the drive.<br>Update of this bit is completely FW-assisted.<br>This bit is called DMA Ready for a DMA ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 4 | 0x0<br>RO/V | **Drive Seek Complete (DSC):** When set, indicates Heads are positioned over the desired cylinder.<br>Update of this bit is completely FW-assisted.<br>This bit is called Service for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 3 | 0x0<br>RO/V | **Data Request (DRQ):** Set when the drive wants to exchange data with the Host via the data register.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 2 | 0x0<br>RO/V | **Corrected Data (CORR):** When set, indicates a correctable read error has occurred.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 1 | 0x0<br>RO/V | **Index (IDX):** This bit is set once per rotation of the medium when the index mark passes under the read/write head.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 0 | 0x0<br>RO/V | **Error (ERR):** When set, indicates an error occurred in the process of executing the current command.<br>The Error Register of the selected device contains the error information.<br>Update of this bit is completely FW-assisted.<br>This bit is called Check for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |

## 54.13.22 IDE Status Register DEV1 (IDESR1)—Offset 7h

This register implements the standard Device 1 Status register in the Primary IDE Command Block. This is read-only. DEV = 1 (i.e. IDEDHIR[4] = 1). CSxE FW writes to this register upon completion of an ATA/ATAPI command. Additionally upon completion of the last data phase of a PIO transfer, HW will load this register with value of the SRAFP field in the IDESRAP register in the CSxE memory space. Host read of this register clears Device 1's interrupt. Please refer to the description of the INTM bit in the IDGCR register in the CSxE memory space for detailed conditions under which interrupt condition shall be cleared. The bit description is for ATA mode. Different name in ATAPI mode, if applicable, is cited as well. Reset condition: CSxE reset with the BSY bit as an exception. Please refer to description of the BSY bit below for a list of BSY bit reset conditions. Non-sticky.

**Type:** I/O Register (Size: 8 bits)

**BAR and Offset:** [IDE_HOST_PCMDIOBAR] + 7h

**Default:** 80h

**IDE_HOST_PCMDIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCMDIOBAR Reference:** [B:0, D:27, F:2] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Busy (BSY):** This bit is set by HW when the IDECR is being written and DEV=1, OR when IDDCR[SRST] bit is asserted by Host, OR host system reset, OR D3 to D0 transition of the IDE function.<br>This bit is cleared by FW write of 0.<br>Please refer to general register description above for special handling for a PIO transfer.<br>When this bit is 1, any other bit in this register is meaningless. |
| 6 | 0x0<br>RO/V | **Drive Ready (DRDY):** When set, indicates drive is ready for command.<br>Update of this bit is completely FW-assisted.<br>This bit is also called Device Ready.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 5 | 0x0<br>RO/V | **Drive Fault (DF):** When set, indicates Error on the drive.<br>Update of this bit is completely FW-assisted.<br>This bit is called DMA Ready for a DMA ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 4 | 0x0<br>RO/V | **Drive Seek Complete (DSC):** When set, indicates Heads are positioned over the desired cylinder.<br>Update of this bit is completely FW-assisted.<br>This bit is called Service for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 3 | 0x0<br>RO/V | **Data Request (DRQ):** Set when the drive wants to exchange data with the Host via the data register.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 2 | 0x0<br>RO/V | **Corrected Data (CORR):** When set, indicates a correctable read error has occurred.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 1 | 0x0<br>RO/V | **Index (IDX):** This bit is set once per rotation of the medium when the index mark passes under the read/write head.<br>Update of this bit is completely FW-assisted.<br>Please refer to general register description above for special handling for a PIO transfer. |
| 0 | 0x0<br>RO/V | **Error (ERR):** When set, indicates an error occurred in the process of executing the current command. The Error Register of the selected device contains the error information.<br>Update of this bit is completely FW-assisted.<br>This bit is called Check for an ATAPI command.<br>Please refer to general register description above for special handling for a PIO transfer. |

## 54.14 Registers in I/O Space—IDE_HOST_PCTLIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_PCTLIOBAR described in Section 54.6.6, "IDE Primary Control Block IO BAR (IDE_HOST_PCTLIOBAR)—Offset 14h" on page 2526.

### 54.14.1 IDE Device Control Register (IDDCR)—Offset 2h

This register implements the standard Device Control register in the Primary IDE Control Block. This is write-only. BE[0] shall be deasserted. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCTLIOBAR] + 2h

**IDE_HOST_PCTLIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCTLIOBAR Reference:** [B:0, D:27, F:2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **Reserved (RSVD1):** Writable by Host, but NO HW effect due to writes. |
| 2 | 0x0<br>WO | **Software Reset (SRST):** 1: both devices shall perform the software reset protocol.<br>0: both devices shall exit software reset. |
| 1 | 0x0<br>WO | **Host Interrupt Disable (nIEN):** 1: disables HW from generating interrupt to the Host.<br>0: enables HW to generate interrupt to the Host. |
| 0 | 0x0<br>WO | **Reserved (RSVD0):** Writable by Host, but NO HW effect due to writes. |

### 54.14.2 IDE Alternate Status Register (IDASR)—Offset 2h

This register implements the standard Alternate Status register in the Primary IDE Control Block. This is read-only. It shadows the standard Status registers in the Primary IDE Command Block. Read of this register does NOT clear the interrupt status bits in the selected Status register. IDESR0 is shadowed when DEV = 0 (i.e. IDEDHIR[4] = 0). IDESR1 is shadowed when DEV = 1 (i.e. IDEDHIR[4] = 1). Reset condition: There is no reset associated with this register because it is NOT a physical register.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_PCTLIOBAR] + 2h

**IDE_HOST_PCTLIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_PCTLIOBAR Reference:** [B:0, D:27, F:2] + 14h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **IDE Alternate Status Register (IDEASR):** Shadows DEV0/DEV1 Status register on Host reads depending on IDEDHIR[4] setting. |

## 54.15 Registers in I/O Space—IDE_HOST_BMIOBAR

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) IDE_HOST_BMIOBAR described in Section 54.6.9, "IDE Bus Master Block IO BAR (IDE_HOST_BMIOBAR)—Offset 20h" on page 2527.

### 54.15.1 IDE Primary Bus Master Command Register (IDEPBMCR)—Offset 0h

This register implements the standard IDE Primary Bus Master Command register in the IDE Bus Master Block. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 0h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** Not implemented. Hardwired to 0. |
| 3 | 0x0<br>RW | **Read or Write Control (RWC):** This bit sets the direction of the bus master transfer.<br>• 0: reads are performed from Host system memory.<br>• 1: writes are performed to Host system memory.<br>This bit shall NOT be changed when the bus master function is active. |
| 2:1 | 0x0<br>RO | **Reserved (RSVD0):** Not implemented. Hardwired to 0. |
| 0 | 0x0<br>RW/V | **Start/Stop Bus Master (SSBM):** Host starts a bus master operation in the device by setting this bit to 1.<br>The device shall only transfer data between the ATA device and Host memory in the bus master operation.<br>Host stops the bus master operation by clearing this bit to 0. All state information will be lost when 0 is written.<br><br>Host CANNOT stop and then resume the bus master operation.<br>When Host clears this bit to 0 while the bus master operation is still active, the bus master operation will be aborted and data transferred from the device may be discarded before being written to host memory.<br>In any event the device should terminate any DMA burst in progress compliant with the ATA protocol being used.<br>If the UDMA protocol is being used, there may be a CRC error report. The ATA device may be in an indeterminate state.<br><br>HW will clear this bit to 0 after the data transfer has been completed, as indicated by satisfaction of either of the following conditions:<br>(1) IDEPBMSR[BMIA] is cleared to 0. OR<br>(2) IDEPBMSR[INT] is set to 1. |

## 54.15.2 IDE Primary Bus Master Device Specific 0 Register (IDEPBMDS0R)—Offset 1h

This register implements the standard IDE Primary Bus Master Device Specific 0 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 1h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Device Specific Data0 (DSD0):** Device Specific. |

## 54.15.3 IDE Primary Bus Master Status Register (IDEPBMSR)—Offset 2h

This register implements the standard IDE Primary Bus Master Status register in the IDE Bus Master Block.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 2h

**Default:** 80h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Simplex Only (SO):** This bit indicates whether or not both bus master channels (primary and secondary) can be operated at the same time.<br>• If the bit is 0, both channels operate independently and can be used at the same time.<br>• If the bit is 1, only one channel may be used at a time.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 6 | 0x0<br>RW/V | **Device 1 DMA Capable (D1DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 1 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance.<br><br>Reset condition: Host reset.<br>Non-sticky. |
| 5 | 0x0<br>RW/V | **Device 0 DMA Capable (D0DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 0 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance.<br><br>Reset condition: Host reset.<br>Non-sticky. |
| 4:3 | 0x0<br>RO | **Reserved (RSVD):** Not implemented. Hardwired to 0. |
| 2 | 0x0<br>RW/1C/V | **Interrupt (INT):** HW shall set this bit to 1 on rising edge detection of the ATA channels interrupt line (INTRQ).<br>Host SW writes 1 to clear this bit. Host SW can use this bit to determine if an ATA device has asserted its interrupt line (INTRQ) (see ATA standard).<br>When this bit is 1, all data MUST have been transferred to the Hosts system memory.<br>The adapter shall not set this bit to 1 until it has flushed all internal data buffers.<br><br>NOTE:<br>When INTRQ is deasserted, HW won't clear this bit atomically.<br>When INTRQ is deasserted, this bit shall be cleared as well by Host SW writing 1 to this bit.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 1 | 0x0<br>RW/1C/V | **Error (ER):** FW will set this bit to 1 when the device encounters an error in transferring data to/from Host memory.<br>The exact error condition is bus specific and can be determined in a bus specific manner.<br>Host SW writes 1 to clear this bit.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RO/V | **Bus Master IDE Active (BMIA):** HW shall set this bit when (1) Host sets the IDEPBMCR[SSBM] bit.<br><br>FW shall clear this bit when<br>(1) The last transfer for a region is performed, where EOT for that region is set in the region descriptor. OR<br>(2) FW detects that Host cleared IDEPBMCR[SSBM] bit as indicated by a value of 1 in proxy bit IDEBMCR[SSBMC] in CSxE root space.<br><br>Before clearing this bit to 0, FW must satisfy the following conditions in addition:<br>(1) All data transferred from the device during the previous bus master command has been transferred to the Hosts system memory, unless the bus master command was aborted. OR<br>(2) All data has been written to the device.<br><br>Reset condition: CSxE reset.<br>Non-sticky. |

## 54.15.4 IDE Primary Bus Master Device Specific 1 Register (IDEPBMDS1R)—Offset 3h

This register implements the standard IDE Primary Bus Master Device Specific 1 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Reset condition: CSxE reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 3h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Device Specific Data1 (DSD1):** Device Specific Data. |

## 54.15.5 IDE Primary Bus Master Descriptor Table Pointer Register Byte 0 (IDEPBMDTPR0)—Offset 4h

This register implements the Byte 0 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 4h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 0 (DTPB0):** Byte 0 of the Descriptor Table pointer.<br>The Descriptor Table shall be DW-aligned.<br>Hence bit[1:0] shall always be 0. |

## 54.15.6 IDE Primary Bus Master Descriptor Table Pointer Register Byte 1 (IDEPBMDTPR1)—Offset 5h

This register implements the Byte 1 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 5h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 1 (DTPB1):** Byte 1 of the Descriptor Table pointer. |

## 54.15.7 IDE Primary Bus Master Descriptor Table Pointer Register Byte 2 (IDEPBMDTPR2)—Offset 6h

This register implements the Byte 2 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 6h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 2 (DTPB2):** Byte 2 of the Descriptor Table pointer. |

## 54.15.8 IDE Primary Bus Master Descriptor Table Pointer Register Byte 3 (IDEPBMDTPR3)—Offset 7h

This register implements the Byte 3 of the standard IDE Primary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the primary channel. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 7h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 3 (DTPB3):** Byte 3 of the Descriptor Table pointer. |

## 54.15.9 IDE Secondary Bus Master Command Register (IDESBMCR)—Offset 8h

This register implements the standard IDE Secondary Bus Master Command register in the IDE Bus Master Block. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 8h

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD1):** Not implemented.<br>Hardwired to 0. |
| 3 | 0x0<br>RW | **Read or Write Control (RWC):** This bit sets the direction of the bus master transfer.<br>• 0: reads are performed from Host system memory.<br>• 1: writes are performed to Host system memory.<br>This bit shall NOT be changed when the bus master function is active. |
| 2:1 | 0x0<br>RO | **Reserved (RSVD0):** Not implemented. Hardwired to 0. |
| 0 | 0x0<br>RW | **Start/Stop Bus Master (SSBM):** Host starts a bus master operation in the device by setting this bit to 1.<br>The device shall only transfer data between the ATA device and Host memory in the bus master operation.<br>Host stops the bus master operation by clearing this bit to 0.<br>All state information will be lost when 0 is written.<br><br>Host CANNOT stop and then resume the bus master operation. |

## 54.15.10 IDE Secondary Bus Master Device Specific 0 Register (IDESBMDS0R)—Offset 9h

This register implements the standard IDE Secondary Bus Master Device Specific 0 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + 9h

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Device Specific Data0 (DSD0):** Device Specific. |

## 54.15.11 IDE Secondary Bus Master Status Register (IDESBMSR)—Offset Ah

This register implements the standard IDE Secondary Bus Master Status register in the IDE Bus Master Block. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset with the SO bit as an exception. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 80h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Ah

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO/V | **Simplex Only (SO):** Value indicates whether both Bus Master Channels can be operated at the same time or not.<br>• 0: Both can be operated independently<br>• 1: Only one can be operated at a time.<br><br>This bit shadows IDEPBMSR[SO].<br><br>Reset condition: CSxE reset.<br>Non-sticky. |
| 6 | 0x0<br>RW | **Device 1 DMA Capable (D1DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 1 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance. |
| 5 | 0x0<br>RW | **Device 0 DMA Capable (D0DC):** This read/write bit is set by device dependent code (BIOS or host software) to indicate that Device 0 for this channel is capable of DMA transfers, and that the device has been initialized for optimum performance. |
| 4:3 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 2 | 0x0<br>RO | **Interrupt (INT):** No functionality implemented.<br>Hardwired to 0. |
| 1 | 0x0<br>RO | **Error (ER):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RO | **Bus Master IDE Active (BMIA):** Not implemented.<br>Hardwired to 0. |

## 54.15.12 IDE Secondary Bus Master Device Specific 1 Register (IDESBMDS1R)—Offset Bh

This register implements the standard IDE Secondary Bus Master Device Specific 1 register in the IDE Bus Master Block. This register is programmed by the Host for any device specific data. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Bh

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Device Specific Data1 (DSD1):** Device Specific Data. |

## 54.15.13 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 0 (IDESBMDTPR0)—Offset Ch

This register implements the Byte 0 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Ch

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 0 (DTPB0):** Byte 0 of the Descriptor Table pointer.<br>The Descriptor Table shall be DW-aligned.<br>Hence bit[1:0] shall always be 0. |

## 54.15.14 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 1 (IDESBMDTPR1)—Offset Dh

This register implements the Byte 1 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Dh

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 1 (DTPB1):** Byte 1 of the Descriptor Table pointer. |

## 54.15.15 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 2 (IDESBMDTPR2)—Offset Eh

This register implements the Byte 2 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Eh

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 2 (DTPB2):** Byte 2 of the Descriptor Table pointer. |

## 54.15.16 IDE Secondary Bus Master Descriptor Table Pointer Register Byte 3 (IDESBMDTPR3)—Offset Fh

This register implements the Byte 3 of the standard IDE Secondary Bus Master Descriptor Table Pointer register in the IDE Bus Master Block. The descriptor table pointer is used for bus master operation in the secondary channel. Usage notes: The Secondary IDE Channel is permanently disabled. Access attributes are maintained to satisfy test requirement. No bus master functionality is implemented behind the register. Reset condition: Host reset. Non-sticky.

**Type:** I/O Register
(Size: 8 bits)

**BAR and Offset:** [IDE_HOST_BMIOBAR] + Fh

**Default:** 00h

**IDE_HOST_BMIOBAR Type:** PCI Configuration Register (Size: 32 bits)
**IDE_HOST_BMIOBAR Reference:** [B:0, D:27, F:2] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Descriptor Table Pointer Byte 3 (DTPB3):** Byte 3 of the Descriptor Table pointer. |

## 54.16 Sideband Registers—KT (Host)

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 54.16.1 KT Host Interrupt Pin Register (KTHIPINR)—Offset 0h

This register provides for BIOS to specify which interrupt pin KT shall use in the Host root space when generating an interrupt in the PCI legacy mode.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xf4] + 0h
**Host Memory Space:** SBREG_BAR + 0xf40000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 7:0 | 0x0<br>RW/O | **Interrupt Pin (IPIN):** BIOS writes to this field once to specify which interrupt pin KT shall use. After written once, this register will become read-only.<br>This field shall be shadowed into the Interrupt Pin register in the PCI header of the KT Host PCI function.<br><br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved<br><br>Reset condition: Host side reset.<br>Non-sticky. |

### 54.16.2 BIOS KT Host PCI Function Disable Register (BKTHDISR)—Offset 4h

This register provides for BIOS to enable or disable the KT Host PCI function.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xf4] + 4h
**Host Memory Space:** SBREG_BAR + 0xf40000 + 4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RW | **Host PCI Function Disable (HDIS):** BIOS sets this bit to disable the KT Host PCI function. BIOS clears this bit to enable the KT Host PCI function.<br><br>Reset condition: Host side reset.<br>Non-sticky. |

## 54.17 Sideband Registers—IDE (Host)

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 54.17.1 IDE Host Interrupt Pin Register (IDEHIPINR)—Offset 0h

This register provides for BIOS to specify which interrupt pin IDE shall use in the Host root space when generating an interrupt in the PCI legacy mode.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xf4] + 0h
**Host Memory Space:** SBREG_BAR + 0xf40000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 7:0 | 0x0<br>RW/O | **Interrupt Pin (IPIN):** BIOS writes to this field once to specify which interrupt pin IDE shall use. After written once, this register will become read-only.<br>This field shall be shadowed into the Interrupt Pin register in the PCI header of the IDE Host PCI function.<br><br>Value Decoding<br>• 00h The function does NOT use an interrupt pin.<br>• 01h INTA<br>• 02h INTB<br>• 03h INTC<br>• 04h INTD<br>• 05h - FFh Reserved<br><br>Reset condition: Host side reset.<br>Non-sticky. |

### 54.17.2 BIOS IDE Host PCI Function Disable Register (BIDEHDISR)—Offset 4h

This register provides for BIOS to enable or disable the IDE Host PCI function.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xf4] + 4h
**Host Memory Space:** SBREG_BAR + 0xf40000 + 4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Not implemented.<br>Hardwired to 0. |
| 0 | 0x0<br>RW | **Host PCI Function Disable (HDIS):** BIOS sets this bit to disable the IDE Host PCI function. BIOS clears this bit to enable the IDE Host PCI function.<br><br>Reset condition: Host side reset.<br>Non-sticky. |

§ §

# 55 eMMC Controller - B0, D28, F0

## 55.1 Introduction and Index

The host-accessible registers for the eMMC Controller are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 55.1.1 Host Memory Space—BAR or BAR1

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 55-1. Summary of Memory Mapped I/O Registers—BAR or BAR1 (Sheet 1 of 3)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| **eMMC Registers** | | | |
| 0 | 4 | 00000000 | "SDMA System Address Register/Argument2 Register (SDMASYSADDR)—Offset 0h" on page 2599 |
| 4 | 2 | 0000 | "BlockSize Register (BLOCKSIZE)—Offset 4h" on page 2599 |
| 6 | 2 | 0000 | "BlockCount Register (BLOCKCOUNT)—Offset 6h" on page 2599 |
| 8 | 4 | 00000000 | "Argument1 Register (ARGUMENT1)—Offset 8h" on page 2600 |
| C | 2 | 0000 | "TransferMode Register (TRANSFERMODE)—Offset Ch" on page 2600 |
| E | 2 | 0000 | "Command Register (COMMAND)—Offset Eh" on page 2600 |
| 10 | 4 | 00000000 | "Response Register (RESPONSE1)—Offset 10h" on page 2601 |
| 14 | 2 | 0000 | "Response Register (RESPONSE2)—Offset 14h" on page 2602 |
| 16 | 2 | 0000 | "Response Register (RESPONSE3)—Offset 16h" on page 2602 |
| 18 | 2 | 0000 | "Response Register (RESPONSE4)—Offset 18h" on page 2602 |
| 1A | 2 | 0000 | "Response Register (RESPONSE5)—Offset 1Ah" on page 2602 |
| 1C | 2 | 0000 | "Response Register (RESPONSE6)—Offset 1Ch" on page 2603 |
| 1E | 2 | 0000 | "Response Register (RESPONSE7)—Offset 1Eh" on page 2603 |
| 20 | 4 | 00000000 | "Buffer DataPort Register (DATAPORT)—Offset 20h" on page 2603 |
| 24 | 4 | 00000000 | "PresentState Register (PRESENTSTATE)—Offset 24h" on page 2604 |
| 28 | 1 | 00 | "HostControl1 Register (HOSTCONTROL1)—Offset 28h" on page 2606 |
| 29 | 1 | 00 | "PowerControl Register (POWERCONTROL)—Offset 29h" on page 2607 |
| 2A | 1 | 80 | "BlockGapControl Register (BLOCKGAPCONTROL)—Offset 2Ah" on page 2607 |
| 2B | 1 | 00 | "Wakeup Control Register (WAKEUPCONTROL)—Offset 2Bh" on page 2608 |
| 2C | 2 | 0000 | "Clock Control Register (CLOCKCONTROL)—Offset 2Ch" on page 2609 |
| 2E | 1 | 00 | "Timeout Control Register (TIMEOUTCONTROL)—Offset 2Eh" on page 2610 |
| 2F | 1 | 00 | "Software Reset Register (SOFTWARERESET)—Offset 2Fh" on page 2610 |
| 30 | 2 | 0000 | "Normal Interrupt Status Register (NORMALINTRSTS)—Offset 30h" on page 2611 |
| 32 | 2 | 0000 | "ErrorInterruptStatus Register (ERRORINTRSTS)—Offset 32h" on page 2612 |
| 34 | 2 | 0000 | "Normal Interrupt Status Enable Register (NORMALINTRSTSENA)—Offset 34h" on page 2613 |
| 36 | 2 | 0000 | "Error Interrupt Status Enable Register (ERRORINTRSTSENA)—Offset 36h" on page 2614 |
| 38 | 2 | 0000 | "Normal Interrupt Signal Enable Register (NORMALINTRSIGENA)—Offset 38h" on page 2615 |
| 3A | 2 | 0000 | "Error Interrupt Signal Enable Register (ERRORINTRSIGENA)—Offset 3Ah" on page 2616 |
| 3C | 2 | 0000 | "Auto CMD12 Error Status Register (AUTOCMDERRSTS)—Offset 3Ch" on page 2617 |
| 3E | 2 | 0000 | "Host Control2 Register (HOSTCONTROL2)—Offset 3Eh" on page 2618 |
| 40 | 8 | 80000007F76F64B2 | "Capabilities Register (CAPABILITIES)—Offset 40h" on page 2619 |

## Table 55-1. Summary of Memory Mapped I/O Registers—BAR or BAR1 (Sheet 2 of 3)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 48 | 8 | 0000000000000000 | "Maximum Current Capabilities Register (MAXCURRENTCAP)—Offset 48h" on page 2621 |
| 50 | 2 | 0000 | "Force Event REGISTER for AUTO CMD Error Status (FORCEEVENTFORAUTOCMDERRORSTATUS)—Offset 50h" on page 2621 |
| 52 | 2 | 0000 | "Force Event Register for Error Interrupt Status (FORCEEVENTFORERRINTSTS)—Offset 52h" on page 2622 |
| 54 | 1 | 00 | "ADMA Error Status Register (ADMAERRSTS)—Offset 54h" on page 2623 |
| 58 | 4 | 00000000 | "ADMA System Address Register0&1 (ADMASYSADDR01)—Offset 58h" on page 2623 |
| 5C | 2 | 0000 | "ADMA System Address Register1 (ADMASYSADDR2)—Offset 5Ch" on page 2623 |
| 5E | 2 | 0000 | "ADMA System Address Register1 (ADMASYSADDR3)—Offset 5Eh" on page 2624 |
| 60 | 2 | 0003 | "Preset Value Register for Initialization (PRESETVALUE0)—Offset 60h" on page 2624 |
| 62 | 2 | 0003 | "Preset Value Register for Default Speed (PRESETVALUE1)—Offset 62h" on page 2625 |
| 64 | 2 | 0003 | "Preset Value Register for High Speed (PRESETVALUE2)—Offset 64h" on page 2625 |
| 66 | 2 | 0003 | "Preset Value Register for SDR12 (PRESETVALUE3)—Offset 66h" on page 2626 |
| 68 | 2 | 0002 | "Preset Value Register for SDR25 (PRESETVALUE4)—Offset 68h" on page 2626 |
| 6A | 2 | 0001 | "Preset Value Register for SDR50 (PRESETVALUE5)—Offset 6Ah" on page 2627 |
| 6C | 2 | 0000 | "Preset Value Register for SDR104 (PRESETVALUE6)—Offset 6Ch" on page 2627 |
| 6E | 2 | 0002 | "Preset Value Register for DDR50 (PRESETVALUE7)—Offset 6Eh" on page 2628 |
| 70 | 4 | 00000000 | "Boot Timeout Control Register (BOOTTIMEOUTCNT)—Offset 70h" on page 2628 |
| 74 | 2 | 0000 | "Preset Value Register for DDR50 (PRESETVALUE8)—Offset 74h" on page 2629 |
| FC | 2 | 0000 | "Slot Interrupt Status Register (SLOTINTRSTS)—Offset FCh" on page 2629 |
| FE | 2 | 1002 | "Host Controller Version Register (HOSTCONTROLLERVER)—Offset FEh" on page 2629 |
| **Extended eMMC Registers (Convergence Layer)** | | | |
| 804 | 4 | 00000800 | "Software Latency Tolerance Reporting (SW_LTR_VAL)—Offset 804h" on page 2630 |
| 808 | 4 | 00000800 | "Auto Latency Tolerance Reporting (AUTO_LTR_VAL)—Offset 808h" on page 2630 |
| 810 | 4 | 00000000 | "Capabilities Bypass Control Register (CAP_BYPS)—Offset 810h" on page 2631 |
| 814 | 4 | 3001EF3C | "eMMC Capabilities Bypass Register1 (CAP_BYPS_REG1)—Offset 814h" on page 2632 |
| 818 | 4 | 04004000 | "eMMC Capabilities Bypass Register2 (CAP_BYPS_REG2)—Offset 818h" on page 2634 |
| 81C | 4 | 00000008 | "D0i3 Control Register (REG_D0i3)—Offset 81Ch" on page 2635 |
| 820 | 4 | 00000400 | "Front End Module Tx Command Path Delay Register (Tx_CMD_DLY)—Offset 820h" on page 2635 |
| 824 | 4 | 00000A18 | "Front End Module Tx Data Path Delay Register1 (Tx_DATA_DLY_1)—Offset 824h" on page 2636 |
| 828 | 4 | 1C1C1C00 | "Front End Module Tx Data Path Delay Register2 (Tx_DATA_DLY_2)—Offset 828h" on page 2637 |

**Table 55-1. Summary of Memory Mapped I/O Registers—BAR or BAR1 (Sheet 3 of 3)**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 82C | 4 | 00000000 | "Front End Module Rx Data Path Delay Register1 (Rx_CMD_DATA_DLY_1)—Offset 82Ch" on page 2638 |
| 830 | 4 | 00000000 | "Front End Module Rx Strobe Path Delay Register (Rx_STROBE_CTRL_PATH)—Offset 830h" on page 2639 |
| 834 | 4 | 00000000 | "Front End Module Rx Data Path Delay Register2 (Rx_CMD_DATA_DLY_2)—Offset 834h" on page 2640 |
| 838 | 4 | 00000001 | "Master DLL Software Control Register (MASTER_DLL)—Offset 838h" on page 2641 |
| 840 | 4 | 00000000 | "Auto Tuning Value (AUTO_TUNING)—Offset 840h" on page 2641 |
| 900 | 4 | 00000000 | "Root Space Select (eMMC_ROOT_SPACE)—Offset 900h" on page 2642 |

## 55.2 Registers in Memory Space—BAR or BAR1

The eMMC operates in either PCI Mode or ACPI Mode. By default all the devices are defined as PCI devices (PCI Mode).

To change a device to an ACPI device (ACPI Mode), it first needs to be enumerated and assigned a BAR. Once this is done, it is possible to configure the device to be an ACPI device. Its address shall be the configured BAR address.

### 55.2.1 SDMA System Address Register/Argument2 Register (SDMASYSADDR)—Offset 0h

This register contains concatenates reg_sdmasysaddrlo and reg_sdmasysaddrhi

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 0h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **sdma_sysaddress:** Refer to reg_sdmasysaddrlo and reg_sdmasysaddrhi |

### 55.2.2 BlockSize Register (BLOCKSIZE)—Offset 4h

This register is used to configure the number of bytes in a data block

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 4h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **RSVD0:** Reserved |
| 14:12 | 0x0<br>RW | **sdma_bufboundary:** This Field specifies DMA Buffer Boundary |
| 11:0 | 0x0<br>RW | **xfer_blocksize:** This Field specifies Block Size for Data Transfers |

### 55.2.3 BlockCount Register (BLOCKCOUNT)—Offset 6h

This register is used to configure the number of data blocks

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 6h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **xfer_blockcount:** This Register specifies Block Size for Data Transfers |

## 55.2.4 Argument1 Register (ARGUMENT1)—Offset 8h

This register contains concatenates argument1lo and argument1hi registers to result SD Command Argument

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 8h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **command_argument1:** refer t oreg_argument1lo and reg_argument1hi. |

## 55.2.5 TransferMode Register (TRANSFERMODE)—Offset Ch

This register is used to control the operations of data transfers

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **RSVD0:** Reserved |
| 5 | 0x0<br>RW | **xfermode_multiblksel:** This bit enables multiple block data transfers. |
| 4 | 0x0<br>RW | **xfermode_dataxferdir:** This bit defines the direction of data transfers. |
| 3:2 | 0x0<br>RW | **xfermode_autocmdena:** This field determines use of auto command functions. |
| 1 | 0x0<br>RW | **xfermode_blkcntena:** This bit is used to enable the Block count register, which is only relevant for multiple block transfers |
| 0 | 0x0<br>RW | **xfermode_dmaenable:** '1' to enable DMA, |

## 55.2.6 Command Register (COMMAND)—Offset Eh

This register is used to program the Command for host controller

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **RSVD0:** Reserved |
| 13:8 | 0x0<br>RW | **command_cmdindex:** This bit shall be set to the command number (CMD0-63, ACMD0-63). |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RW | **command_cmdtype:** This bit is used for select different command types. |
| 5 | 0x0<br>RW | **command_datapresent:** This bit is used to checking whether data present or not. |
| 4 | 0x0<br>RW | **command_indexchkena:** This bit is used to enable/disable Command Index checking. |
| 3 | 0x0<br>RW | **command_crcchkena:** This bit is used to enable/disable CRC checking. |
| 2 | 0x0<br>RO | **RSVD1:** Reserved |
| 1:0 | 0x0<br>RW | **command_responsetype:** Response Type Select. |

## 55.2.7 Response Register (RESPONSE1)—Offset 10h

This register is used to store responses from SD Cards (concatinates response0 and response1 registers)

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 10h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.8 Response Register (RESPONSE2)—Offset 14h

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [BAR or BAR1] + 14h

**Default:** 0000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.9 Response Register (RESPONSE3)—Offset 16h

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [BAR or BAR1] + 16h

**Default:** 0000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.10 Response Register (RESPONSE4)—Offset 18h

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [BAR or BAR1] + 18h

**Default:** 0000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.11 Response Register (RESPONSE5)—Offset 1Ah

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [BAR or BAR1] + 1Ah

**Default:** 0000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.12 Response Register (RESPONSE6)—Offset 1Ch

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 1Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.13 Response Register (RESPONSE7)—Offset 1Eh

This register is used to store responses from SD Cards

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 1Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO | **command_response:** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit |

## 55.2.14 Buffer DataPort Register (DATAPORT)—Offset 20h

This register is used to access internal buffer

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 20h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **sdhcdmactrl_piobufrddata:** The Host Controller Buffer can be accessed through this 32-bit Data Port Register. |

## 55.2.15 PresentState Register (PRESENTSTATE)—Offset 24h

The Host Driver can get status of the Host Controller from this 32-bit read-only register

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 24h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:29 | 0x0<br>RO | **RSVD0:** Reserved |
| 28 | 0x0<br>RO | **sdif_dat7in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 27 | 0x0<br>RO | **sdif_dat6in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 26 | 0x0<br>RO | **sdif_dat5in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 25 | 0x0<br>RO | **sdif_dat4in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 24 | 0x0<br>RO | **sdif_cmdin_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 23 | 0x0<br>RO | **sdif_dat3in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 22 | 0x0<br>RO | **sdif_dat2in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 21 | 0x0<br>RO | **sdif_dat1in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 20 | 0x0<br>RO | **sdif_dat0in_dsync:** This status is used to check DAT line level to recover from errors, and for debugging. |
| 19 | 0x0<br>RO | **sdif_wp_dsync:** The Write Protect Switch is supported for memory and combo cards. |
| 18 | 0x0<br>RO | **sdif_cd_n_dsync:** This bit reflects the inverse value of the SDCD# pin. |
| 17 | 0x0<br>RO | **sdhccarddet_statestable_dsync:** This bit is used for testing. |
| 16 | 0x0<br>RO | **sdhccarddet_inserted_dsync:** This bit indicates whether a card has been inserted. |
| 15:12 | 0x0<br>RO | **RSVD1:** Reserved |
| 11 | 0x0<br>RO | **sdhcdmactrl_piobufrdena:** This status is used for non-DMA read transfers.<br>This read only flag indicates that valid data exists in the host side buffer status. |
| 10 | 0x0<br>RO | **sdhcdmactrl_piobufwrena:** This status is used for non-DMA write transfers.<br>This read only flag indicates if space is available for write data. |
| 9 | 0x0<br>RO | **sdhcdmactrl_rdxferactive:** This status is used for detecting completion of a read transfer. |
| 8 | 0x0<br>RO | **sdhcdmactrl_wrxferactive:** This status indicates a write transfer is active. |
| 7:4 | 0x0<br>RO | **RSVD2:** Reserved |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RO | **sdhcsdctrl_retuningreq_dsync:** Host Controller may request Host Driver to execute re-tuning sequence by setting this bit. |
| 2 | 0x0<br>RO | **sdhcdmactrl_datalineactive:** This bit indicates whether one of the DAT line on SD bus is in use. |
| 1 | 0x0<br>RO | **presentstate_inhibitdat:** This status bit is generated if either the DAT Line Active or the Read transfer Active is set to 1. |
| 0 | 0x0<br>RO | **presentstate_inhibitcmd:** If this bit is 0, it indicates the CMD line is not in use and the HC can issue a SD command using the CMD line.<br>This bit is set immediately after the Command register (00Fh) is written.<br>This bit is cleared when the command response is received. |

## 55.2.16 HostControl1 Register (HOSTCONTROL1)—Offset 28h

This register is used to program DMA modes, LED Control, Data Transfer Width, High Speed Enable, Card detect test level and signal selection.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 28h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **hostctrl1_cdsigselect:** This bit selects source for card detection. |
| 6 | 0x0<br>RW | **hostctrl1_cdtestlevel:** This bit is used for indicating whether card is inserted or not. |
| 5 | 0x0<br>RW | **hostctrl1_extdatawidth:** This bit controls 8-bit bus width mode for embedded device. |
| 4:3 | 0x0<br>RW | **hostctrl1_dmaselect:** One of supported DMA modes can be selected. |
| 2 | 0x0<br>RW | **hostctrl1_highspeedena:** This bit is used for setting speed mode. |
| 1 | 0x0<br>RW | **hostctrl1_datawidth:** This bit selects the data width of the HC. |
| 0 | 0x0<br>RW | **hostctrl1_ledcontrol:** This bit is used to caution the user not to remove the card while the SD card is being accessed. |

## 55.2.17 PowerControl Register (POWERCONTROL)—Offset 29h

This register is used to program the SD Bus power and voltage level

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 29h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **RSVD0:** Reserved |
| 4 | 0x0<br>RW | **emmc_hwreset:** Hardware reset signal is generated for eMMC card. |
| 3:1 | 0x0<br>RW | **pwrctrl_sdbusvoltage:** By setting these bits, the HD selects the voltage level for the SD card. |
| 0 | 0x0<br>RW | **pwrctrl_sdbuspower:** This bit is used detect whether power is on or off. |

## 55.2.18 BlockGapControl Register (BLOCKGAPCONTROL)—Offset 2Ah

This register is used to program the block gap request, read wait control and interrupt at block gap

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 80h

**BAR and Offset:** [BAR or BAR1] + 2Ah

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **blkgapctrl_bootackena:** To check for the boot acknowledge in boot operation. |
| 6 | 0x0<br>RW | **blkgapctrl_altbootmode:** To start boot code access in alternative mode. |
| 5 | 0x0<br>RW | **blkgapctrl_bootenable:** To start boot code access.<br>• '0' To stop boot code access<br>• '1' To start boot code access |
| 4 | 0x0<br>RW | **blkgapctrl_spimode:** SPI mode enable bit.<br>• '0' SD Mode<br>• '1' SPI Mode |
| 3 | 0x0<br>RW | **blkgapctrl_interrupt:** This bit is valid only in 4-bit mode of the SDIO card and selects a sample point in the interrupt cycle. |
| 2 | 0x0<br>RW | **blkgapctrl_rdwaitctrl:** The read wait function is optional for SDIO cards. |
| 1 | 0x0<br>RW | **blkgapctrl_continue:** This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request. |
| 0 | 0x0<br>RW | **blkgapctrl_stopatblkgap:** This bit is used to stop executing a transaction at the next block gap for non- DMA,SDMA and ADMA transfers. |

## 55.2.19 Wakeup Control Register (WAKEUPCONTROL)—Offset 2Bh

This register is used to program the wakeup functionality

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 2Bh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **wkupctrl_cardremoval:** This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. |
| 1 | 0x0<br>RW | **wkupctrl_cardinsertion:** This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register. |
| 0 | 0x0<br>RW | **wkupctrl_cardinterrupt:** This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register. |

## 55.2.20 Clock Control Register (CLOCKCONTROL)—Offset 2Ch

This register is used to program the Clock frequency select, generator select, Clock enable, Internal Clock state fields

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**BAR and Offset:** [BAR or BAR1] + 2Ch

**Default:** 0000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **clkctrl_sdclkfreqsel:** This register is used to select the frequency of the SDCLK pin. |
| 7:6 | 0x0<br>RW | **clkctrl_sdclkfreqsel_upperbits:** Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select |
| 5 | 0x0<br>RW | **clkctrl_clkgensel:** This bit is used to select the clock generator mode in SDCLK Frequency Select. |
| 4:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **clkctrl_sdclkena:** This bit enables/disables SD Clock.<br>• '0' Disable<br>• '1' Enable |
| 1 | 0x0<br>RO | **sdhcclkgen_intclkstable_dsync:** This bit is set to 1 when SD clock is stable after writing to Internal Clock Enable in this register to 1. |
| 0 | 0x0<br>RW | **clkctrl_intclkena:** This bit enables setting of internal clock.<br>• '0' Stop<br>• '1' Oscillate |

## 55.2.21 Timeout Control Register (TIMEOUTCONTROL)—Offset 2Eh

The register sets the Data Timeout counter value

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 2Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **RSVD0:** Reserved |
| 3:0 | 0x0<br>RW | **timeout_ctrvalue:** This value determines the interval by which DAT line time-outs are detected. |

## 55.2.22 Software Reset Register (SOFTWARERESET)—Offset 2Fh

This register is used to program the software reset for data, command and for all.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 2Fh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RW | **swreset_for_dat:** Only part of data circuit is reset.<br>• '0' Work<br>• '1' Reset |
| 1 | 0x0<br>RW | **swreset_for_cmd:** Only part of command circuit is reset.<br>• '0' Work<br>• '1' reset. |
| 0 | 0x0<br>RW | **swreset_for_all:** If this bit is set to 1, the SD card shall reset itself and must be re initialized by the HD. |

## 55.2.23 Normal Interrupt Status Register (NORMALINTRSTS)—Offset 30h

This register gives the status of all the interrupts.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 30h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **reg_errorintrsts:** This bit is set If any of the bits in the Error Interrupt Status Register are set. |
| 14 | 0x0<br>RW | **normalintrsts_bootcomplete:** This status is set if the boot operation gets terminated. |
| 13 | 0x0<br>RW | **normalintrsts_rcvbootack:** This status is set if the boot acknowledge is received from device. |
| 12 | 0x0<br>RO | **normalintrsts_retuningevent:** This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. |
| 11 | 0x0<br>RO | **normalintrsts_intc:** This status is set if INT_C is enabled and INT_C# pin is in low level.<br>It is cleared by resetting the INT_C interrupt factor. |
| 10 | 0x0<br>RO | **normalintrsts_intb:** This status is set if INT_B is enabled and INT_B# pin is in low level.<br>It is cleared by resetting |
| 9 | 0x0<br>RO | **normalintrsts_inta:** This status is set if INT_A is enabled and INT_A# pin is in low level.<br>It is cleared by resetting |
| 8 | 0x0<br>RO | **normalintrsts_cardintsts:** This status is set to generate Card Interrupt.<br>• '0' No Card Interrupt<br>• '1' Generate Card Interrupt |
| 7 | 0x0<br>RW | **normalintrsts_cardremsts:** This status is set if the Card Inserted in the Present State register changes from 1 to 0. |
| 6 | 0x0<br>RW | **normalintrsts_cardinssts:** This status is set if the Card Inserted in the Present State register changes from 0 to 1. |
| 5 | 0x0<br>RW | **normalintrsts_bufrdready:** This status is set if the Buffer Read Enable changes from 0 to 1. |
| 4 | 0x0<br>RW | **normalintrsts_bufwrready:** This status is set if the Buffer Write Enable changes from 0 to 1. |
| 3 | 0x0<br>RW | **normalintrsts_dmainterrupt:** This status is set if the HC detects the Host DMA Buffer Boundary in the Block Size register. |
| 2 | 0x0<br>RW | **normalintrsts_blkgapevent:** If the Stop At Block Gap Request in the BlockGap Control Register is set, this bit is set. |
| 1 | 0x0<br>RW | **normalintrsts_xfercomplete:** This bit is set when a read / write transaction is completed. |
| 0 | 0x0<br>RW | **normalintrsts_cmdcomplete:** This bit is set when we get the end bit of the command response. |

## 55.2.24 ErrorInterruptStatus Register (ERRORINTRSTS)—Offset 32h

This register gives the status of the error interrupts.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 32h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **RSVD0:** Reserved |
| 12 | 0x0<br>RW | **errorintrsts_hosterror:** Occurs when detecting Host ERROR |
| 11:10 | 0x0<br>RO | **RSVD1:** Reserved |
| 9 | 0x0<br>RW | **errorintrsts_admaerror:** This bit is set when the Host Controller detects errors during ADMA based data transfer. |
| 8 | 0x0<br>RW | **errorintrsts_autocmderror:** This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register has changed from 0 to 1. |
| 7 | 0x0<br>RW | **errorintrsts_currlimiterror:** If this bit is 1, it means that the HC is not supplying power to SD card due to some failure.<br>Reading 0 means that the HC is supplying power and no error has occurred. |
| 6 | 0x0<br>RW | **errorintrsts_dataendbiterror:** Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status. |
| 5 | 0x0<br>RW | **errorintrsts_datacrcerror:** Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than 010. |
| 4 | 0x0<br>RW | **errorintrsts_datatimeouterror:** This status is set if Data Timeout error occurs. |
| 3 | 0x0<br>RW | **errorintrsts_cmdindexerror:** Occurs if a Command Index error occurs in the Command Response. |
| 2 | 0x0<br>RW | **errorintrsts_cmdendbiterror:** Occurs when detecting that the end bit of a command response is 0. |
| 1 | 0x0<br>RW | **errorintrsts_cmdcrcerror:** This bit is set when Command CRC Error is generated.<br>• '0' No Error<br>• '1' CRC Error |
| 0 | 0x0<br>RW | **errorintrsts_cmdtimeouterror:** This bit is set if no response is returned within 64 SDCLK cycles from the end bit of the command. |

## 55.2.25 Normal Interrupt Status Enable Register (NORMALINTRSTSENA)—Offset 34h

This register is used to enable the normal interrupt status register fields.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 34h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **normalintrsts_enableregbit15:** The HC shall control error Interrupts using the Error Interrupt Status Enable register. |
| 14 | 0x0<br>RW | **normalintrsts_enableregbit14:** This status is set if the boot operation gets terminated. |
| 13 | 0x0<br>RW | **normalintrsts_enableregbit13:** This status is set if the boot acknowledge is received from device. |
| 12 | 0x0<br>RW | **normalintrsts_enableregbit12:** This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. |
| 11 | 0x0<br>RW | **normalintrsts_enableregbit11:** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. |
| 10 | 0x0<br>RW | **normalintrsts_enableregbit10:** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. |
| 9 | 0x0<br>RW | **normalintrsts_enableregbit9:** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System. |
| 8 | 0x0<br>RW | **sdhcregset_cardintstsena:** If this bit is set to 0, the HC shall clear Interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1. |
| 7 | 0x0<br>RW | **sdhcregset_cardremstsena:** This status is set if the Card Inserted in the Present State register changes from 1 to 0. |
| 6 | 0x0<br>RW | **sdhcregset_cardinsstsena:** This status is set if the Card Inserted in the Present State register changes from 0 to 1. |
| 5 | 0x0<br>RW | **normalintrsts_enableregbit5:** This status is set if the Buffer Read Enable changes from 0 to 1. |
| 4 | 0x0<br>RW | **normalintrsts_enableregbit4:** This status is set if the Buffer Write Enable changes from 0 to 1. |
| 3 | 0x0<br>RW | **normalintrsts_enableregbit3:** This status is set if the HC detects the Host DMA Buffer Boundary in the Block Size regiser. |
| 2 | 0x0<br>RW | **normalintrsts_enableregbit2:** If the Stop At Block Gap Request in the BlockGap Control Register is set, this bit is set. |
| 1 | 0x0<br>RW | **normalintrsts_enableregbit1:** This bit is set when a read / write transaction is completed. |
| 0 | 0x0<br>RW | **normalintrsts_enableregbit0:** This bit is set when we get the end bit of the command response. |

## 55.2.26 Error Interrupt Status Enable Register (ERRORINTRSTSENA)—Offset 36h

This register is used to enable the Error Interrupt Status register fields

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 36h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **RSVD0:** Reserved |
| 12 | 0x0<br>RO | **errorintrsts_enableregbit12:** Occurs when detecting ERROR in m_hresp(dma transaction) |
| 11 | 0x0<br>RO | **RSVD1:** Reserved |
| 10 | 0x0<br>RW | **errorintrsts_enableregbit10:** This status is set if INT_B is enabled and INT_B# pin is in low level. It is cleared by resetting |
| 9 | 0x0<br>RW | **errorintrsts_enableregbit9:** This bit is set when the Host Controller detects errors during ADMA based data transfer. |
| 8 | 0x0<br>RW | **errorintrsts_enableregbit8:** This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register has changed from 0 to 1. |
| 7 | 0x0<br>RW | **errorintrsts_enableregbit7:** If this bit is 1, it means that the HC is not supplying power to SD card due to some failure.<br>Reading 0 means that the HC is supplying power and no error has occurred. |
| 6 | 0x0<br>RW | **errorintrsts_enableregbit6:** Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status. |
| 5 | 0x0<br>RW | **errorintrsts_enableregbit5:** Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than 010. |
| 4 | 0x0<br>RW | **errorintrsts_enableregbit4:** This status is set if Data Timeout error occurs. |
| 3 | 0x0<br>RW | **errorintrsts_enableregbit3:** Occurs if a Command Index error occurs in the Command Response. |
| 2 | 0x0<br>RW | **errorintrsts_enableregbit2:** Occurs when detecting that the end bit of a command response is 0. |
| 1 | 0x0<br>RW | **errorintrsts_enableregbit1:** This bit is set when Command CRC Error is generated. |
| 0 | 0x0<br>RW | **errorintrsts_enableregbit0:** This bit is set if no response is returned within 64 SDCLK cycles from the end bit of the command. |

## 55.2.27 Normal Interrupt Signal Enable Register (NORMALINTRSIGENA)—Offset 38h

This register is used to enable the Normal Interrupt Signal register.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 38h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **normalintrsts_enableregbit15:** The HD shall control error Interrupts using the Error Interrupt Signal Enable register. |
| 14 | 0x0<br>RW | **normalintrsts_enableregbit14:** Boot Terminate Interrupt Signal Enable |
| 13 | 0x0<br>RW | **normalintrsts_enableregbit13:** Boot ack rcv Signal Enable |
| 12 | 0x0<br>RW | **normalintrsts_enableregbit12:** Re-Tuning Event Signal Enable |
| 11 | 0x0<br>RW | **normalintrsts_enableregbit11:** INT_C Signal Enable |
| 10 | 0x0<br>RW | **normalintrsts_enableregbit10:** INT_B Signal Enable |
| 9 | 0x0<br>RW | **normalintrsts_enableregbit9:** INT_A Signal Enable |
| 8 | 0x0<br>RW | **sdhcregset_cardintstsena:** Card Interrupt Signal Enable |
| 7 | 0x0<br>RW | **sdhcregset_cardremstsena:** Card Removal Signal Enable |
| 6 | 0x0<br>RW | **sdhcregset_cardinsstsena:** Card Insertion Signal Enable |
| 5 | 0x0<br>RW | **normalintrsts_enableregbit5:** Buffer Read Ready Signal Enable |
| 4 | 0x0<br>RW | **normalintrsts_enableregbit4:** Buffer Write Ready Signal Enable |
| 3 | 0x0<br>RW | **normalintrsts_enableregbit3:** DMA Interrupt Signal Enable |
| 2 | 0x0<br>RW | **normalintrsts_enableregbit2:** Block Gap Event Signal Enable |
| 1 | 0x0<br>RW | **normalintrsts_enableregbit1:** Transfer Complete Signal Enable |
| 0 | 0x0<br>RW | **normalintrsts_enableregbit0:** Command Complete Signal Enable |

## 55.2.28 Error Interrupt Signal Enable Register (ERRORINTRSIGENA)—Offset 3Ah

This register is used to enable Error Interrupt Signal register.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 3Ah

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **RSVD0:** Reserved |
| 12 | 0x0<br>RO | **errorintrsig_enableregbit12:** Target Response Error Signal Enable |
| 11 | 0x0<br>RO | **RSVD1:** Reserved |
| 10 | 0x0<br>RW | **errorintrsig_enableregbit10:** Tuning Error Signal Enable |
| 9 | 0x0<br>RW | **errorintrsig_enableregbit9:** ADMA Error Signal Enable |
| 8 | 0x0<br>RW | **errorintrsig_enableregbit8:** Auto CMD Error Signal Enable |
| 7 | 0x0<br>RW | **errorintrsig_enableregbit7:** Current Limit Error Signal Enable |
| 6 | 0x0<br>RW | **errorintrsig_enableregbit6:** Data End Bit Error Signal Enable |
| 5 | 0x0<br>RW | **errorintrsig_enableregbit5:** Data CRC Error Signal Enable |
| 4 | 0x0<br>RW | **errorintrsig_enableregbit4:** Data Timeout Error Signal Enable |
| 3 | 0x0<br>RW | **errorintrsig_enableregbit3:** Command Index Error Signal Enable |
| 2 | 0x0<br>RW | **errorintrsig_enableregbit2:** Command End Bit Error Signal Enable |
| 1 | 0x0<br>RW | **errorintrsig_enableregbit1:** Command CRC Error Signal Enable |
| 0 | 0x0<br>RW | **errorintrsig_enableregbit0:** Command Timeout Error Signal Enable |

## 55.2.29 Auto CMD12 Error Status Register (AUTOCMDERRSTS)—Offset 3Ch

This register is used to indicate CMD12 response error of Auto CMD12 and CMD23 response error of Auto CMD 23.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 3Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7 | 0x0<br>RO | **autocmderrsts_nexterror:** Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error(D04- D01) in this register.<br>This bit is set to 0 when Auto CMD Error is generated by Auto CMD23 |
| 6:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RO | **autocmderrsts_indexerror:** Occurs if the Command Index error occurs in response to a command. |
| 3 | 0x0<br>RO | **autocmderrsts_endbiterror:** Occurs when detecting that the end bit of command response is 0. |
| 2 | 0x0<br>RO | **autocmderrsts_crcerror:** Occurs when detecting a CRC error in the command response. |
| 1 | 0x0<br>RO | **autocmderrsts_timeouterror:** Occurs if the no response is returned within 64 SDCLK cycles from the end bit of the command. |
| 0 | 0x0<br>RO | **autocmderrsts_notexecerror:** Setting this bit to 1 means the HC cannot issue Auto CMD12 to stop memory multiple block transfer due to some error. |

## 55.2.30 Host Control2 Register (HOSTCONTROL2)—Offset 3Eh

This register is used to program UHS Select Mode, UHS Select Mode, Driver Strength Select, Execute Tuning, Sampling Clock Select, Asynchronous Interrupt Enable and Preset value enable.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 3Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **hostctrl2_presetvalueenable:** This bit enables the functions defined in the Preset Value registers. |
| 14 | 0x0<br>RW | **hostctrl2_asynchintrenable:** This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register. |
| 13:10 | 0x0<br>RO | **RSVD0:** Reserved |
| 9 | 0x0<br>RW | **hostctrl2_driverstrength_bit2:** This is the programmed Drive Strength output and Bit[2] of the sdhccore_drivestrength value. |
| 8 | 0x0<br>RO | **RSVD1:** Reserved |
| 7 | 0x0<br>RW | **hostctrl2_samplingclkselect:** This bit is set by tuning procedure when Execute Tuning is cleared. |
| 6 | 0x0<br>RW | **hostctrl2_executetuning:** This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed. |
| 5:4 | 0x0<br>RW | **hostctrl2_driverstrength:** Host Controller output driver in 1.8V signaling is selected by this bit. |
| 3 | 0x0<br>RW | **hostctrl2_1p8vsignallingena:** This bit controls voltage regulator for I/O cell. |
| 2:0 | 0x0<br>RW | **hostctrl2_uhsmodeselect:** This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1. |

## 55.2.31 Capabilities Register (CAPABILITIES)—Offset 40h

This register provides the host driver with information specific to the host controller implementation.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 80000007F76F64B2h

**BAR and Offset:** [BAR or BAR1] + 40h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63 | 0x1<br>RO | **corecfg_hs400support:** This field indicates whether HS400 is supported or not. |
| 62:58 | 0x0<br>RO | **rsvd0:** Reserved Field |
| 57 | 0x0<br>RO | **corecfg_spiblkmode:** This field indicates whether SPI Block Mode is supported or not. |
| 56 | 0x0<br>RO | **corecfg_spisupport:** This field indicates whether SPI Mode is supported or not. |
| 55:48 | 0x0<br>RO | **corecfg_clockmultiplier:** This field indicates clock multiplier value of programmable clock generator. |
| 47:46 | 0x0<br>RO | **corecfg_retuningmodes:** This field defines the re-tuning capability of a Host Controller. |
| 45 | 0x0<br>RO | **corecfg_tuningforsdr50:** If this bit is set to 1, this Host Controller requires tuning to operate SDR50. |
| 44 | 0x0<br>RO | **rsvd1:** Reserved Field |
| 43:40 | 0x0<br>RO | **corecfg_retuningtimercnt:** This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. |
| 39 | 0x0<br>RO | **corecfg_type4support:** This bit indicates support of Type 4 Driver. |
| 38 | 0x0<br>RO | **corecfg_ddriversupport:** This bit indicates support of Driver Type D for 1.8 Signaling. |
| 37 | 0x0<br>RO | **corecfg_cdriversupport:** This bit indicates support of Driver Type C for 1.8 Signaling. |
| 36 | 0x0<br>RO | **corecfg_adriversupport:** This bit indicates support of Driver Type A for 1.8 Signaling. |
| 35 | 0x0<br>RO | **rsvd2:** Reserved Field |
| 34 | 0x1<br>RO | **corecfg_ddr50support:** This bit indicates whether DDR50 is supported. |
| 33 | 0x1<br>RO | **corecfg_sdr104support:** This bit indicates whether SDR104 is supported.SDR104 requires tuning. |
| 32 | 0x1<br>RO | **corecfg_sdr50support:** This bit indicates whether SDR50 is supported. |
| 31:30 | 0x3<br>RO | **corecfg_slottype:** This field indicates usage of a slot by a specific Host System |
| 29 | 0x1<br>RO | **corecfg_asynchintrsupport:** Asynchronous Interrupt Support |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 28 | 0x1<br>RO | **corecfg_64bitsupport:** This bit indicates whether the HC supports 64bit System Bus |
| 27 | 0x0<br>RO | **rsvd3:** Reserved Field |
| 26 | 0x1<br>RO | **corecfg_1p8voltsupport:** This bit indicates whether the HC supports 1.8V. |
| 25 | 0x1<br>RO | **corecfg_3p0voltsupport:** This bit indicates whether the HC supports 3.0V. |
| 24 | 0x1<br>RO | **corecfg_3p3voltsupport:** This bit indicates whether the HC supports 3.3V. |
| 23 | 0x0<br>RO | **corecfg_suspressupport:** This bit indicates whether the HC supports Suspend/Resume functionality. |
| 22 | 0x1<br>RO | **corecfg_sdmasupport:** This bit indicates whether the HC is capable of using DMA to transfer data between system memory and the HC directly. |
| 21 | 0x1<br>RO | **corecfg_highspeedsupport:** This bit indicates whether the HC and the Host System support High Speed mode. |
| 20 | 0x0<br>RO | **rsvd4:** Reserved Field |
| 19 | 0x1<br>RO | **corecfg_adma2support:** '0'ADMA2 Not Supported |
| 18 | 0x1<br>RO | **corecfg_8bitsupport:** This bit indicates whether the Host Controller is capable of using 8-bit bus width mode. |
| 17:16 | 0x3<br>RO | **corecfg_maxblklength:** This value indicates the maximum block size that the HD can read and write to the buffer in the HC. |
| 15:8 | 0x64<br>RO | **corecfg_baseclkfreq:** 6-bit Base Clock Frequency |
| 7 | 0x1<br>RO | **corecfg_timeoutclkunit:** This bit shows the unit of base clock frequency used to detect Data Timeout Error. |
| 6 | 0x0<br>RO | **rsvd5:** Reserved Field |
| 5:0 | 0x32<br>RO | **corecfg_timeoutclkfreq:** This bit shows the base clock frequency used to detect Data Timeout Error. |

## 55.2.32 Maximum Current Capabilities Register (MAXCURRENTCAP)—Offset 48h

This register indicates maximum current capability for each voltage.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Default:** 0000000000000000h

**BAR and Offset:** [BAR or BAR1] + 48h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x0<br>RO | **corecfg_maxcurrent1p8v:** Maximum Current for 1.8V |
| 15:8 | 0x0<br>RO | **corecfg_maxcurrent3p0v:** Maximum Current for 3.0V |
| 7:0 | 0x0<br>RO | **corecfg_maxcurrent3p3v:** Maximum Current for 3.3V |

## 55.2.33 Force Event REGISTER for AUTO CMD Error Status (FORCEEVENTFORAUTOCMDERRORSTATUS)—Offset 50h

This register is not physically implemented, rather it is an address where Auto CMD Error Status register can be written.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 50h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7 | 0x0<br>RO | **forcecmdnotissuedbyautocmd12err:** Force Event for Command Not Issued by AUTO CMD12 Error |
| 6:5 | 0x0<br>RO | **RSVD1:** Reserved |
| 4 | 0x0<br>RO | **forceautocmdindexerr:** Force Event for AUTO CMD Index Error |
| 3 | 0x0<br>RO | **forceautocmdendbiterr:** Force Event for AUTO CMD End Bit Error |
| 2 | 0x0<br>RO | **forceautocmdcrcerr:** Force Event for AUTO CMD Timeout Error |
| 1 | 0x0<br>RO | **forceautocmdtimeouterr:** Force Event for AUTO CMD Timeout Error |
| 0 | 0x0<br>RO | **forceautocmdnotexec:** Force Event for AUTO CMD12 Not Executed |

## 55.2.34 Force Event Register for Error Interrupt Status (FORCEEVENTFORERRINTSTS)—Offset 52h

This register is not physically implemented, rather it is an address where Error Interrupt Status register can be written.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 52h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **forcetuningerr:** Force Event for Tuning Error |
| 9 | 0x0<br>RO | **forceadmaerr:** Force Event for ADMA Error<br>• '0' no interrupt<br>• '1' interrupt generated |
| 8 | 0x0<br>RO | **forceautocmderr:** Force Event for Auto CMD Error<br>• '0' no interrupt<br>• '1' interrupt generated |
| 7 | 0x0<br>RO | **forcecurrlimerr:** Force Event for Current Limit Error<br>• '0' no interrupt<br>• '1' interrupt generated |
| 6 | 0x0<br>RO | **forcedatendbiterr:** Force Event for Data End Bit Error.<br>• '0' no interrupt<br>• '1' interrupt generated |
| 5 | 0x0<br>RO | **forcedatcrcerr:** Force Event for Data CRC Error |
| 4 | 0x0<br>RO | **forcedattimeouterr:** Force Event for Data Timeout Error<br>• '0' no interrupt<br>• '1' interrupt generated |
| 3 | 0x0<br>RO | **forcecmdindexerr:** Force Event for Command Index Error |
| 2 | 0x0<br>RO | **forcecmdendbiterr:** Force Event for Command End Bit Error<br>• '0' no interrupt<br>• '1' interrupt generated |
| 1 | 0x0<br>RO | **forcecmdcrcerr:** Force Event for Command CRC Error |
| 0 | 0x0<br>RO | **forcecmdtimeouterr:** Force Event for CMD Timeout Error<br>• '0' No interrupt<br>• '1' interrupt generated |

## 55.2.35 ADMA Error Status Register (ADMAERRSTS)—Offset 54h

When the ADMA Error interrupt occurs, this register holds the ADMA State in ADMA Error States field and ADMA System Address holds address around the error descriptor.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [BAR or BAR1] + 54h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **RSVD0:** Reserved |
| 2 | 0x0<br>RO | **admaerrsts_admalenmismatcherr:** ADMA Length Mismatch Error |
| 1:0 | 0x0<br>RO | **admaerrsts_admaerrorstate:** This field indicates the state of ADMA when error is occurred during ADMA data transfer. |

## 55.2.36 ADMA System Address Register0&1 (ADMASYSADDR01)—Offset 58h

This register contains the physical address used for ADMA data transfer.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 58h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **adma_sysaddress0:** This register holds byte address of executing command of the Descriptor table. |

## 55.2.37 ADMA System Address Register1 (ADMASYSADDR2)—Offset 5Ch

This register contains the physical address used for ADMA data transfer.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 5Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **adma_sysaddress2:** This register holds byte address of executing command of the Descriptor table. |

## 55.2.38 ADMA System Address Register1 (ADMASYSADDR3)—Offset 5Eh

This register contains the physical address used for ADMA data transfer.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 5Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW | **adma_sysaddress3:** This register holds byte address of executing command of the Descriptor table. |

## 55.2.39 Preset Value Register for Initialization (PRESETVALUE0)—Offset 60h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0003h

**BAR and Offset:** [BAR or BAR1] + 60h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x3<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.40 Preset Value Register for Default Speed (PRESETVALUE1)—Offset 62h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for Default Speed.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0003h

**BAR and Offset:** [BAR or BAR1] + 62h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x3<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.41 Preset Value Register for High Speed (PRESETVALUE2)—Offset 64h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for High Speed.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0003h

**BAR and Offset:** [BAR or BAR1] + 64h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x3<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.42 Preset Value Register for SDR12 (PRESETVALUE3)—Offset 66h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for SDR12.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0003h

**BAR and Offset:** [BAR or BAR1] + 66h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x3<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.43 Preset Value Register for SDR25 (PRESETVALUE4)—Offset 68h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value SDR25.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0002h

**BAR and Offset:** [BAR or BAR1] + 68h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x2<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.44 Preset Value Register for SDR50 (PRESETVALUE5)—Offset 6Ah

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for SDR50.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0001h

**BAR and Offset:** [BAR or BAR1] + 6Ah

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x1<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.45 Preset Value Register for SDR104 (PRESETVALUE6)—Offset 6Ch

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for SDR 104.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 6Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x0<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.46 Preset Value Register for DDR50 (PRESETVALUE7)—Offset 6Eh

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for DDR50.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0002h

**BAR and Offset:** [BAR or BAR1] + 6Eh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x2<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.47 Boot Timeout Control Register (BOOTTIMEOUTCNT)—Offset 70h

This is used to program the boot timeout value counter.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 70h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **boot_timeoutcnt:** This value determines the interval by which DAT line time-outs are detected during boot operation for eMMC4.4 card. |

## 55.2.48 Preset Value Register for DDR50 (PRESETVALUE8)—Offset 74h

This register is used to read the SDCLK Frequency Select Value, Clock Generator Select Value, and Driver Strength Select Value for DDR50.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + 74h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **DriverStrengthSelectValue:** Driver Strength Select Value |
| 13:11 | 0x0<br>RO | **RSVD0:** Reserved |
| 10 | 0x0<br>RO | **ClockGeneratorSelectValue:** This bit is effective when Host Controller supports programmable clock |
| 9:0 | 0x0<br>RO | **SDCLKFrequencySelectValue:** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system. |

## 55.2.49 Slot Interrupt Status Register (SLOTINTRSTS)—Offset FCh

This register is used to read the interrupt signal for each slot.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [BAR or BAR1] + FCh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RO | **RSVD0:** Reserved |
| 7:0 | 0x0<br>RO | **sdhchostif_slotintrsts:** These status bits indicate the logical OR of Interrupt signal and Wakeup signal for each slot. |

## 55.2.50 Host Controller Version Register (HOSTCONTROLLERVER)—Offset FEh

This register is used to read the vendor version number and specification version number.

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 1002h

**BAR and Offset:** [BAR or BAR1] + FEh

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x10<br>RO | **SDHC_VENVERNUM:** This status is reserved for the vendor version number. |
| 7:0 | 0x2<br>RO | **SpecificationVersionNumber:** This status indicates the Host Controller Spec. Version.<br>The upper and lower 4-bits indicate the version. |

## 55.2.51 Software Latency Tolerance Reporting (SW_LTR_VAL)—Offset 804h

Sets the Software Latency Tolerance Reporting fields.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000800h

**BAR and Offset:** [BAR or BAR1] + 804h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15 | 0x0<br>RW | **Snoop_Requirment:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely).<br>If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time. |
| 14:13 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 12:10 | 0x2<br>RW | **Snoop_latency_scale:** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale (1us -) 32ms total span) only.<br>Writes to this CSR which don't match those values will be dropped completely, next read will return previous value. |
| 9:0 | 0x0<br>RW | **Snoop_value:** 10-bit latency value |

## 55.2.52 Auto Latency Tolerance Reporting (AUTO_LTR_VAL)—Offset 808h

Sets the Auto Latency Tolerance Reporting fields.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000800h

**BAR and Offset:** [BAR or BAR1] + 808h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15 | 0x0<br>RW | **Snoop_Requirment:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely).<br>If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time. |
| 14:13 | 0x0<br>RO | **Reserved_low:** Reserved_low |
| 12:10 | 0x2<br>RW | **Snoop_latency_scale:** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale (1us -) 32ms total span) only.<br>Writes to this CSR which don't match those values will be dropped completely, next read will return previous value. |
| 9:0 | 0x0<br>RW | **Snoop_value:** 10-bit latency value |

## 55.2.53 Capabilities Bypass Control Register (CAP_BYPS)—Offset 810h

Capabilities Bypass Control register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 810h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7:0 | 0x0<br>RW | **Enable_Cap_Bypass:** 8h5A Enable Capabilities Bypass.<br>All other Capabilities Bypass Disable (using default values). |

## 55.2.54 eMMC Capabilities Bypass Register1 (CAP_BYPS_REG1)—Offset 814h

eMMC Capabilities Bypass register 1

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 3001EF3Ch

**BAR and Offset:** [BAR or BAR1] + 814h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 29 | 0x1<br>RW | **hs400_support:**<br>• 1: HS400 Mode Supported.<br>• 0: HS400 Mode NOT Supported |
| 28 | 0x1<br>RW | **timeout_clock_unit:**<br>• 1b1 - to Select MHz Clock<br>• 1b0 - to Select KHz Clock |
| 27:22 | 0x200<br>RW | **timeout_clock_freq:** Timeout clock frequency |
| 21 | 0x0<br>RW | **SPI_mode_support:** SPI Mode Support<br>• 1b1 SPI Mode Supported<br>• 1b0 SPI Mode Not Supported |
| 20:17 | 0x0<br>RW | **timer_count:** Timer Count for Re-Tuning This is the Timer Count for Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 4b0 disables Re-Tuning Timer |
| 16 | 0x1<br>RW | **tuning_for_SDR50:** Use Tuning for SDR50<br>• 1b1 Use Tuning<br>• 1b0 Don’t use Tuning |
| 15 | 0x1<br>RW | **ddr50_support:**<br>• 1: DDR50 Mode Supported.<br>• 0: DDR50 Mode NOT Supported |
| 14 | 0x1<br>RW | **sdr104_support:**<br>• 1: SDR104 Mode Supported.<br>• 0: SDR104 Mode NOT Supported |
| 13 | 0x1<br>RW | **sdr50_support:**<br>• 1: SDR50 Mode Supported.<br>• 0: SDR50 Mode NOT Supported |
| 12:11 | 0x1<br>RW | **Slot_Type:**<br>• 00 - Removable SD Card Slot.<br>• 01 - Embedded Slot for One Device.<br>• 10 - Shared Bus Slot.<br>• 11 - Reserved |
| 10 | 0x1<br>RW | **Async_Interrupt_Support:**<br>• 1: Asynchronous Interrupt Supported.<br>• 0: Asynchronous Interrupt NOT Supported |
| 9 | 0x1<br>RW | **Sys_Addr_64bit_Support:**<br>• 1 - Core supports 64-bit System Address Bus.<br>• 0 - Core supports only 32-bit System Address Bus |
| 8 | 0x1<br>RW | **Voltage_Support_1_8V:**<br>• 1: 1.8V Supported.<br>• 0: 1.8V NOT Supported |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Voltage_Support_3V:**<br>• 1: 3.0V Supported.<br>• 0: 3.0V NOT Supported |
| 6 | 0x0<br>RW | **Voltage_Support_3_3V:**<br>• 1: 3.3V Supported.<br>• 0: 3.3V NOT Supported |
| 5 | 0x1<br>RW | **Suspend_Resume_Support:**<br>• 1: Suspend/Resume Supported.<br>• 0: Suspend/Resume NOT Supported |
| 4 | 0x1<br>RW | **SDMA_Support:**<br>• 1: SDMA mode Supported.<br>• 0: SDMA mode NOT Supported |
| 3 | 0x1<br>RW | **High_Speed_Support:**<br>• 1: HIGH_SPEED mode Supported.<br>• 0: HIGH_SPEED mode NOT Supported |
| 2 | 0x1<br>RW | **ADMA2_Support:**<br>• 1: ADMA2 mode Supported.<br>• 0: ADMA2 mode NOT Supported |
| 1:0 | 0x0<br>RW | **Max_Burst_Length:** Maximum Block Length supported by the Core/Device.<br>• 00: 512 (Bytes).<br>• 01: 1024.<br>• 10: 2048.<br>• 11: Reserved |

## 55.2.55 eMMC Capabilities Bypass Register2 (CAP_BYPS_REG2)—Offset 818h

eMMC Capabilities Bypass register 2

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 04004000h

**BAR and Offset:** [BAR or BAR1] + 818h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 26:21 | 0x20<br>RW | **tuning_count_val:** Tuning Count Value Configures the Number of Taps (Phases) of the rxclk_in that is supported.<br>The Tuning State machine uses this information to select one of the Taps (Phases) of the rxclk_in during the Tuning Procedure. |
| 20 | 0x0<br>RW | **tuning_dis:** Disable the 1.5x Tuning count when calculating total tuning count.<br>The internal tuning count will be set to the corecfg_Tuningcount when this signal is asserted |
| 19 | 0x0<br>RW | **driver_type_4:** Driver Type 4 Support<br>• 1b1 Supported<br>• 1b0 NOT Supported |
| 18 | 0x0<br>RW | **driver_type_D:** Driver Type D Support<br>• 1b1 Supported<br>• 1b0 NOT Supported |
| 17 | 0x0<br>RW | **driver_type_C:** Driver Type C Support<br>• 1b1 Supported<br>• 1b0 NOT Supported |
| 16 | 0x0<br>RW | **driver_type_A:** Driver Type A Support<br>• 1b1 Supported<br>• 1b0 NOT Supported |
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14 | 0x1<br>RW | **support_8_bit_embedded:** 8-bit Support for Embedded Device<br>• 1b1 Supported<br>• 1b0 NOT Supported |
| 13:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7:0 | 0x0<br>RW | **base_sd_clock:** Base Clock Frequency for SD Clock |

## 55.2.56 D0i3 Control Register (REG_D0i3)—Offset 81Ch

D0i3 Control register

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000008h

**BAR and Offset:** [BAR or BAR1] + 81Ch

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x1<br>RW/1C | **RestoreRequired:** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost.<br>SW clears the bit by writing a 1. This bit will be set on initial power up. |
| 2 | 0x0<br>RW | **D0i3:** SW sets this bit to 1 to move the IP into the D0i3 state.<br>Writing this bit to 0 will return the IP to the fully active D0 state (D0i0). |
| 1 | 0x0<br>RO | **Interrupt_Request:** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command.<br>SW must clear or set this on each write to this register. Not supported in SCS! |
| 0 | 0x0<br>RO | **Cmd_In_Progress:** HW sets this bit on a 1-)0 or 0-)1 transition of bit [2].<br>While set, the other bits in this register are not valid and it is illegal for SW to write to any bit in this register.<br>When clear all the other bits in the register are valid and SW may write to any bit.<br>If Interrupt Request bit [1] was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles |

## 55.2.57 Front End Module Tx Command Path Delay Register (Tx_CMD_DLY)—Offset 820h

Front end module Tx Command Path Delay register

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000400h

**BAR and Offset:** [BAR or BAR1] + 820h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:8 | 0x4<br>RW | **ddr_mode:** Tx CMD Delay (DDR Mode). 0 39 Select number of active delay elements.<br>Each = 125pSec. 40 127 Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x0<br>RW | **sdr_mode:** Tx CMD Delay (SDR Mode). 0 39 Select number of active delay elements.<br>Each = 125pSec. 40 127 - Reserved |

## 55.2.58 Front End Module Tx Data Path Delay Register1 (Tx_DATA_DLY_1)—Offset 824h

Front end module Tx data Path Delay register 1

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000A18h

**BAR and Offset:** [BAR or BAR1] + 824h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:8 | 0xa<br>RW | **hs400_mode:** Tx Data Delay (HS400 Mode).<br>• 0-78 - Select number of active delay elements. Each = 125pSec.<br>• 79-127 - Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x18<br>RW | **sdr104_hs200:** Tx Data Delay (SDR104/HS200 Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |

## 55.2.59 Front End Module Tx Data Path Delay Register2 (Tx_DATA_DLY_2)—Offset 828h

Front end module Tx data Path Delay register 2

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 1C1C1C00h

**BAR and Offset:** [BAR or BAR1] + 828h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:24 | 0x1c<br>RW | **sdr50_mode:** Tx Data Delay (SDR50 Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |
| 23 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 22:16 | 0x1c<br>RW | **ddr50_mode:** Tx Data Delay (DDR50 Mode).<br>• 0-78 - Select number of active delay elements. Each = 125pSec.<br>• 79-127 - Reserved |
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:8 | 0x1c<br>RW | **sdr25_hs50_mode:** Tx Data Delay (SDR25/HS50 Mode<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x0<br>RW | **sdr12_comp_mode:** Tx Data Delay (SDR12/Compatibility Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |

## 55.2.60 Front End Module Rx Data Path Delay Register1 (Rx_CMD_DATA_DLY_1)—Offset 82Ch

Front end module Rx data Path Delay register 1

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BAR or BAR1] + 82Ch

**Default:** 00000000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:24 | 0x0<br>RW | **sdr50_mode:** Rx CMD + Data Delay (SDR50 Mode).<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |
| 23 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 22:16 | 0x0<br>RW | **ddr50_mode:** Rx CMD + Data Delay (DDR50 Mode).<br>• 0-78 - Select number of active delay elements. Each = 125pSec.<br>• 79-127 - Reserved |
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:8 | 0x0<br>RW | **sdr25_hs50:** Rx CMD + Data Delay (SDR25/HS50 Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x0<br>RW | **sdr12_comp:** Rx CMD + Data Delay (SDR12/Compatibility Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |

## 55.2.61 Front End Module Rx Strobe Path Delay Register (Rx_STROBE_CTRL_PATH)—Offset 830h

Front end module Rx strobe Path Delay register

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 830h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 16 | 0x0<br>RW | **auto_tuning:** Enable Auto Tuning for HS400 Strobe Path<br>• 0 Auto Tuning Disabled<br>• 1 Auto Tuning Enabled |
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:8 | 0x0<br>RW | **hs400_mode1:** Rx Strobe Delay DLL 1(HS400 Mode)<br>• 0-39 - Select number of active delay elements. Each = 125pSec<br>• 40-63 - Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x0<br>RW | **hs400_mode2:** Rx Strobe Delay DLL 2(HS400 Mode)<br>• 0-39 - Select number of active delay elements. Each = 125pSec<br>• 40-63 - Reserved |

## 55.2.62 Front End Module Rx Data Path Delay Register2 (Rx_CMD_DATA_DLY_2)—Offset 834h

Front end module Rx data Path Delay register 2

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BAR or BAR1] + 834h

**Default:** 00000000h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 17:16 | 0x0<br>RW | **clk_source:** Clock Source for Rx Path<br>• 00 - Rx Clock after Output Buffer<br>• 01 - Rx Clock before Output Buffer<br>• 10 - Automatic Selection based on Working mode (HS 200 before buffer, all others after buffer)<br>• 11 - Reserved |
| 15:14 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 13:8 | 0x0<br>RW | **path_pll:** Rx Path PLL #3 Delay value For Auto Tuning Mode<br>• 0-39 - Select the required delay, as a multiple of 125pSec.<br>• 40-63 - Reserved |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6:0 | 0x0<br>RW | **cmd_data_sdr104_hs200:** Rx CMD + Data Delay (SDR104/HS200 Mode)<br>• 0-79 - Select the required delay, as a multiple of 125pSec.<br>• 80-127 - Reserved |

## 55.2.63 Master DLL Software Control Register (MASTER_DLL)—Offset 838h

Master DLL Software control register

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000001h

**BAR and Offset:** [BAR or BAR1] + 838h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 24 | 0x0<br>RW | **SW_reset_dll:** SW reset for Master DLL<br>• 0 No SW Reset for Master DLL<br>• 1 Force Reset for Master DLL |
| 23 | 0x0<br>RO/V | **DLL_lock:** Master DLL Lock Indication |
| 22:8 | 0x0<br>RW | **coarse_code:** Set coarse code to DLL. (only valid when Software control is Enabled) |
| 7:4 | 0x0<br>RW | **fine_code:** Set fine code to DLL. (only valid when Software control is Enabled) |
| 3 | 0x0<br>RO/V | **less:** Phase Detection Less Indication |
| 2 | 0x0<br>RO/V | **more:** Phase Detector More Indication |
| 1 | 0x0<br>RW | **Master_DLL_Software_Ctrl:** Master DLL Software Ctrl.<br>• 0 Master DLL Automatic Control (SW Control Disabled).<br>• 1 Master DLL Software Control Enabled |
| 0 | 0x1<br>RW | **Ctrl_of_Mst_DLL_Ref_Clk:** Ctrl of Master DLL Ref Clock.<br>• 0 - Clock is Disabled.<br>• 1 - Clock is Enabled |

## 55.2.64 Auto Tuning Value (AUTO_TUNING)—Offset 840h

Auto Tuning Value

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 840h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 4:0 | 0x0<br>RO/V | **Auto_tuning_val:** Reserved. |

## 55.2.65 Root Space Select (eMMC_ROOT_SPACE)—Offset 900h

Root space select to indicate which root space the IP will output its upstream accesses

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BAR or BAR1] + 900h

**BAR or BAR1 Type:** PCI Configuration Register (Size: 64 bits)
**BAR or BAR1 Reference:** [B:0, D:28, F:0] + 10h or 18h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 0 | 0x0<br>RO | **root_space:** Root Space Select Selects the Root Space in which the IP output its upstream accesses<br>• 0 Root Space 0<br>• 1 Root Space 1 |

§ §

# 56 LPC Controller - B0, D31, F0

## 56.1 Introduction and Index

The host-accessible registers for the LPC Controller are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 56.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 0. The offset addresses are listed.

**Table 56-1. Summary of PCI Configuration Registers—0/31/0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 80 | 2 | 0000 | "I/O Decode Ranges (IOD)—Offset 80h" on page 2645 |
| 82 | 2 | 0000 | "I/O Enables (IOE)—Offset 82h" on page 2646 |
| 84 | 4 | 00000000 | "LPC Generic I/O Range #1 (LGIR1)—Offset 84h" on page 2647 |
| 88 | 4 | 00000000 | "LPC Generic I/O Range #2 (LGIR2)—Offset 88h" on page 2648 |
| 8C | 4 | 00000000 | "LPC Generic I/O Range #3 (LGIR3)—Offset 8Ch" on page 2649 |
| 90 | 4 | 00000000 | "LPC Generic I/O Range #4 (LGIR4)—Offset 90h" on page 2650 |
| 94 | 4 | 00000000 | "USB Legacy Keyboard/Mouse Control (ULKMC)—Offset 94h" on page 2651 |
| 98 | 4 | 00000000 | "LPC Generic Memory Range (LGMR)—Offset 98h" on page 2653 |
| D0 | 4 | 00112233 | "FWH ID Select #1 (FS1)—Offset D0h" on page 2653 |
| D4 | 2 | 4567 | "FWH ID Select #2 (FS2)—Offset D4h" on page 2654 |
| D8 | 2 | FFCF | "BIOS Decode Enable (BDE)—Offset D8h" on page 2655 |
| DC | 1 | 20 | "BIOS Control (BC)—Offset DCh" on page 2657 |
| E0 | 4 | 00000000 | "PCI Clock Control (PCCTL)—Offset E0h" on page 2659 |

## 56.2 Registers in Configuration Space

### 56.2.1 I/O Decode Ranges (IOD)—Offset 80h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:0] + 80h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 12 | 0x0<br>RW | **FDD Range (FDD):** The following table describes which range to decode for the FDD Port:<br>Bits Decode Range<br>• 0 3F0h - 3F5h, 3F7h (Primary).<br>• 1 370h - 375h, 377h (Secondary). |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 9:8 | 0x0<br>RW | **LPT Range (LPT):** The following table describes which range to decode for the LPT Port:<br>Bits Decode Range<br>• 00 378h - 37Fh and 778h - 77Fh.<br>• 01 278h - 27Fh (port 279h is read only) and 678h - 67Fh.<br>• 10 3BCh - 3BEh and 7BCh - 7BEh.<br>• 11 Reserved. |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 6:4 | 0x0<br>RW | **ComB Range (CB):** The following table describes which range to decode for the COMB Port:<br>Bits Decode Range<br>• 000 3F8h - 3FFh (COM 1)<br>• 001 2F8h - 2FFh (COM 2)<br>• 010 220h - 227h<br>• 011 228h - 22Fh<br>• 100 238h - 23Fh<br>• 101 2E8h - 2EFh (COM 4)<br>• 110 338h - 33Fh<br>• 111 3E8h - 3EFh (COM 3) |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 2:0 | 0x0<br>RW | **ComA Range (CA):** The following table describes which range to decode for the COMA Port:<br>Bits Decode Range<br>• 000 3F8h - 3FFh (COM 1)<br>• 001 2F8h - 2FFh (COM 2)<br>• 010 220h - 227h<br>• 011 228h - 22Fh<br>• 100 238h - 23Fh<br>• 101 2E8h - 2EFh (COM 4)<br>• 110 338h - 33Fh<br>• 111 3E8h - 3EFh (COM 3) |

## 56.2.2 I/O Enables (IOE)—Offset 82h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:0] + 82h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 13 | 0x0<br>RW | **Microcontroller Enable #2 (ME2):** Enables decoding of I/O locations 4Eh and 4Fh to LPC. |
| 12 | 0x0<br>RW | **SuperI/O Enable (SE):** Enables decoding of I/O locations 2Eh and 2Fh to LPC. |
| 11 | 0x0<br>RW | **Microcontroller Enable #1 (ME1):** Enables decoding of I/O locations 62h and 66h to LPC. |
| 10 | 0x0<br>RW | **Keyboard Enable (KE):** Enables decoding of the keyboard I/O locations 60h and 64h to LPC. |
| 9 | 0x0<br>RW | **High Gameport Enable (HGE):** Enables decoding of the I/O locations 208h to 20Fh to LPC. |
| 8 | 0x0<br>RW | **Low Gameport Enable (LGE):** Enables decoding of the I/O locations 200h to 207h to LPC.<br><br>**Power Well:** Core |
| 7:4 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 3 | 0x0<br>RW | **Floppy Drive Enable (FDE):** Enables decoding of the FDD range to LPC.<br>Range is selected by IOD.FDE |
| 2 | 0x0<br>RW | **Parallel Port Enable (PPE):** Enables decoding of the LPT range to LPC.<br>Range is selected by IOD.LPT. |
| 1 | 0x0<br>RW | **Com Port B Enable (CBE):** Enables decoding of the COMB range to LPC.<br>Range is selected IOD.CB. |
| 0 | 0x0<br>RW | **Com Port A Enable (CAE):** Enables decoding of the COMA range to LPC.<br>Range is selected IOD.CA. |

### 56.2.3 LPC Generic I/O Range #1 (LGIR1)—Offset 84h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 23:18 | 0x0<br>RW | **Address [7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match.<br>The corresponding bit in the Address field, below, is ignored.<br>The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 15:2 | 0x0<br>RW | **Address [15:2] (Address_15_2):** DWord-aligned address.<br><br>***Note:*** The SoC does not provide decode down to the word or byte level. |
| 1 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 0 | 0x0<br>RW | **LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC. |

## 56.2.4 LPC Generic I/O Range #2 (LGIR2)—Offset 88h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 23:18 | 0x0<br>RW | **Address [7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match.<br>The corresponding bit in the Address field, below, is ignored.<br>The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 15:2 | 0x0<br>RW | **Address [15:2] (Address_15_2):** DWord-aligned address.<br><br>***Note:*** The SoC does not provide decode down to the word or byte level. |
| 1 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 0 | 0x0<br>RW | **LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC. |

## 56.2.5 LPC Generic I/O Range #3 (LGIR3)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 8Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 23:18 | 0x0<br>RW | **Address [7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match.<br>The corresponding bit in the Address field, below, is ignored.<br>The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 15:2 | 0x0<br>RW | **Address [15:2] (Address_15_2):** DWord-aligned address.<br><br>***Note:*** The SoC does not provide decode down to the word or byte level. |
| 1 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 0 | 0x0<br>RW | **LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC. |

## 56.2.6 LPC Generic I/O Range #4 (LGIR4)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 23:18 | 0x0<br>RW | **Address [7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match.<br>The corresponding bit in the Address field, below, is ignored.<br>The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size. |
| 17:16 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 15:2 | 0x0<br>RW | **Address [15:2] (Address_15_2):** DWord-aligned address.<br><br>***Note:*** The SoC does not provide decode down to the word or byte level. |
| 1 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 0 | 0x0<br>RW | **LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC. |

## 56.2.7 USB Legacy Keyboard/Mouse Control (ULKMC)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15 | 0x0<br>RW/1C | **SMI Caused by End of Pass-through (SMIBYENDPS):** Indicates if the event occurred.<br><br>***Note:*** Even if the corresponding enable bit is not set in the Bit 7, then this bit will still be active.<br><br>It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>Writing a 1 to this bit will clear the latch. |
| 14 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 13 | 0x0<br>RW | **Reserved (RSVD_2):** Reserved. |
| 12 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 11 | 0x0<br>RW/1C | **SMI Caused by Port 64 Write (TRAPBY64W):** Indicates if the event occurred.<br>***Note:*** Even if the corresponding enable bit is not set in the Bit 3, then this bit will still be active.<br>It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>Writing a 1 to this bit will clear the latch.<br><br>***Note:*** The A20Gate Pass-Through Logic allows specific port 64h Writes to complete without setting this bit. |
| 10 | 0x0<br>RW/1C | **SMI Caused by Port 64 Read (TRAPBY64R):** Indicates if the event occurred.<br>***Note:*** Even if the corresponding enable bit is not set in the Bit 2, then this bit will still be active.<br>It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>Writing a 1 to this bit will clear the latch. |
| 9 | 0x0<br>RW/1C | **SMI Caused by Port 60 Write (TRAPBY60W):** Indicates if the event occurred.<br><br>***Note:*** Even if the corresponding enable bit is not set in the Bit 1, then this bit will still be active.<br><br>It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>Writing a 1 to this bit will clear the latch.<br><br>***Note:*** The A20Gate Pass-Through Logic allows specific port 60h Writes to complete without setting this bit. |
| 8 | 0x0<br>RW/1C | **SMI Caused by Port 60 Read (TRAPBY60R):** Indicates if the event occurred.<br><br>***Note:*** Even if the corresponding enable bit is not set in the Bit 0, then this bit will still be active.<br><br>It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#.<br>Writing a 1 to this bit will clear the latch. |
| 7 | 0x0<br>RW | **SMI at End of Pass-through Enable (SMIATENDPS):** May need to cause SMI at the end of a pass-through.<br>Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later. |
| 6 | 0x0<br>RO | **Pass Through State (PSTATE):** This read-only bit indicates that the state machine is in the middle of an A20GATE pass-through sequence.<br>If software needs to reset this bit, it should set Bit 5 0. |
| 5 | 0x0<br>RW | **A20Gate Pass-Through Enable (A20PASSEN):** When enabled, allows A20GATE sequence Pass-Through function (see details below in Section 20.14). When enabled, a specific cycle sequence involving writes to port 60h and port 64h does not result in the setting of the SMI status bits.SMI# will not be generated, even if the various enable bits are set. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW | **Reserved (RSVD_4):** Reserved. |
| 3 | 0x0<br>RW | **SMI on Port 64 Writes Enable (s64WEN):** When set, a 1 in bit 11 will cause an SMI event. |
| 2 | 0x0<br>RW | **SMI on Port 64 Reads Enable (s64REN):** When set, a 1 in bit 10 will cause an SMI event. |
| 1 | 0x0<br>RW | **SMI on Port 60 Writes Enable (s60WEN):** When set, a 1 in bit 9 will cause an SMI event. |
| 0 | 0x0<br>RW | **SMI on Port 60 Reads Enable (s60REN):** When set, a 1 in bit 8 will cause an SMI event. |

## 56.2.8 LPC Generic Memory Range (LGMR)—Offset 98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Memory Address [31:16] (MA_31_16):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC as standard LPC Memory Cycle if enabled. |
| 15:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 0 | 0x0<br>RW | **LPC Memory Range Decode Enable (LMRD_En):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC. |

## 56.2.9 FWH ID Select #1 (FS1)—Offset D0h

This register contains the IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + D0h

**Default:** 00112233h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **F8-FF IDSEL (IF8):** IDSEL to use in FWH cycle for range enabled by BDE.EF8. |
| 27:24 | 0x0<br>RW | **F0-F7 IDSEL (IF0):** IDSEL to use in FWH cycle for range enabled by BDE.EF0. |
| 23:20 | 0x1<br>RW | **E8-EF IDSEL (IE8):** IDSEL to use in FWH cycle for range enabled by BDE.EE8. |
| 19:16 | 0x1<br>RW | **E0-E7 IDSEL (IE0):** IDSEL to use in FWH cycle for range enabled by BDE.EE0. |
| 15:12 | 0x2<br>RW | **D8-DF IDSEL (ID8):** IDSEL to use in FWH cycle for range enabled by BDE.ED8. |
| 11:8 | 0x2<br>RW | **D0-D7 IDSEL (ID0):** IDSEL to use in FWH cycle for range enabled by BDE.ED0. |
| 7:4 | 0x3<br>RW | **C8-CF IDSEL (IC8):** IDSEL to use in FWH cycle for range enabled by BDE.EC8. |
| 3:0 | 0x3<br>RW | **C0-C7 IDSEL (IC0):** IDSEL to use in FWH cycle for range enabled by BDE.EC0. |

## 56.2.10 FWH ID Select #2 (FS2)—Offset D4h

This register contains the additional IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:0] + D4h

**Default:** 4567h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x4<br>RW | **70-7F IDSEL (I70):** IDSEL to use in FWH cycle for range enabled by BDE.E70. |
| 11:8 | 0x5<br>RW | **60-6F IDSEL (I60):** IDSEL to use in FWH cycle for range enabled by BDE.E60. |
| 7:4 | 0x6<br>RW | **50-5F IDSEL (I50):** IDSEL to use in FWH cycle for range enabled by BDE.E50. |
| 3:0 | 0x7<br>RW | **40-4F IDSEL (I40):** IDSEL to use in FWH cycle for range enabled by BDE.E40. |

## 56.2.11 BIOS Decode Enable (BDE)—Offset D8h

Note that this register effects the BIOS decode regardless of whether the BIOS is resident on LPC or SPI. The concept of Feature Space does not apply to SPI-based flash. The SoC simply decodes these ranges as memory accesses when enabled for the SPI flash interface.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:0] + D8h

**Default:** FFCFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x1<br>RO | **F8-FF Enable (EF8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFF80000h - FFFFFFFFh.<br>• Feature space: FFB80000h - FFBFFFFFh. |
| 14 | 0x1<br>RW | **F0-F8 Enable (EF0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFF00000h - FFF7FFFFh.<br>• Feature space: FFB00000h - FFB7FFFFh. |
| 13 | 0x1<br>RW | **E8-EF Enable (EE8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFE80000h - FFEFFFFFh.<br>• Feature space: FFA80000h - FFAFFFFFh. |
| 12 | 0x1<br>RW | **E0-E8 Enable (EE0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFE00000h - FFE7FFFFh.<br>• Feature Space: FFA00000h - FFA7FFFFh. |
| 11 | 0x1<br>RW | **D8-DF Enable (ED8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFD80000h - FFDFFFFFh.<br>• Feature space: FF980000h - FF9FFFFFh. |
| 10 | 0x1<br>RW | **D0-D7 Enable (ED0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFD00000h - FFD7FFFFh.<br>• Feature space: FF900000h - FF97FFFFh. |
| 9 | 0x1<br>RW | **C8-CF Enable (EC8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFC80000h - FFCFFFFFh.<br>• Feature space: FF880000h - FF8FFFFFh. |
| 8 | 0x1<br>RW | **C0-C7 Enable (EC0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFC00000h - FFC7FFFFh.<br>• Feature space: FF800000h - FF87FFFFh. |
| 7 | 0x1<br>RW | **Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h - FFFFFh.<br>***Note:*** Decode for the BIOS legacy F segment is enabled by the LFE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit. |
| 6 | 0x1<br>RW | **Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h - EFFFFh.<br>***Note:*** Decode for the BIOS legacy E segment is enabled by the LEE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 3 | 0x1<br>RW | **70-7F Enable (E70):** Enables decoding of 1MB of the following BIOS range:<br>• Data space: FF700000h - FF7FFFFFh.<br>• Feature space: FF300000h - FF3FFFFFh. |
| 2 | 0x1<br>RW | **60-6F Enable (E60):** Enables decoding of 1MB of the following BIOS range:<br>• Data space: FF600000h - FF6FFFFFh.<br>• Feature Space: FF200000h - FF2FFFFFh. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x1<br>RW | **50-5F Enable (E50):** Enables decoding of 1MB of the following BIOS range:<br>• Data space: FF500000h - FF5FFFFFh.<br>• Feature space: FF100000h - FF1FFFFFh. |
| 0 | 0x1<br>RW | **40-4F Enable (E40):** Enables decoding of 1MB of the following BIOS range:<br>• Data space: FF400000h - FF4FFFFFh.<br>• Feature space: FF000000h - FF0FFFFFh. |

## 56.2.12 BIOS Control (BC)—Offset DCh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:0] + DCh

**Default:** 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1L | **BIOS Interface Lock-Down (BILD):** When set, prevents BC.TS and BC.BBS from being changed. This bit can only be written from 0 to 1 once.<br>***Note:*** BIOS: This bit is not backed up in the RTC well. This bit should also be set in the BUC register in the RTC device to record the last state of this value following a cold reset. |
| 6 | 0x0<br>RW/L | **Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. For the default, Functional Strap section of Signal Description chapter for details.<br>Bits Description<br>• 0 SPI.<br>• 1 LPC.<br>When SPI or LPC is selected, the range that is decoded is further qualified by other configuration bits described in the respective sections.<br>The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down is not set.<br>***Note:*** Software must not change this bit since the SoC does not support dynamic switching between the SPI and LPC interface it is chosen at boot time by a strap. |
| 5 | 0x1<br>RW/L | **Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until SMM sets the InSMM.STS bit.<br>If this bit [5] is set, then WPD must be a 1 and InSMM.STS must be 1 also.<br>If this bit [5] is clear, then BIOS is writable based only on WPD = 1 and the InSMM.STS is a don't care. |
| 4 | 0x0<br>RO | **Top Swap (TS):** When set, the SoC will invert either A16, A17, or A18 for cycles going to the BIOS space (but not the feature space). When cleared, the SoC will not invert A16.<br>If booting from SPI, then the BOOT_BLOCK_SIZE soft strap determines if A16, A17, or A18 should be inverted if Top Swap is enabled.<br>*If the SoC is strapped for to enable Top-Swap Override (GPIO[3] is sampled as height (1) within 400 ns of the rising edge (0 -> 1) of the active-low resume reset signal RSMRST_N), then this bit cannot be cleared by software. The strap jumper should be removed (GPIO[3] set to low(0)) and the system rebooted.<br>BIOS Note:<br>1. This bit provides a read-only path to view the state of the Top Swap strap. It is backed up and driven from the RTC well. BIOS will need to program the corresponding register in the RTC Controller (in RTC well), which will be reflected in this register.<br>2. The Register portion of the Top Swap is lockable by the BIOS Interface Lockdown Bit (BC.BILD) but unlockable by SPI Flash Protected Range and Top Swap Override (uCode.PRR_TS_OVR). |
| 3:2 | 0x0<br>RW | **SPI Read Configuration (SRC):** This 2-bit field controls two policies related to BIOS reads on the SPI interface:<br>Bit 3- Prefetch Enable<br>Bit 2- Cache Disable<br>Settings are summarized below:<br>Bits 3:2 Description<br>• 00 - No prefetching, but caching enabled. Direct Memory reads load the read buffer cache with valid data, allowing repeated reads to the same range to complete quickly<br>• 01 - No prefetching and no caching. One-to-one correspondence of host BIOS reads to SPI cycles. This value can be used to invalidate the cache.<br>• 10 - Prefetching and Caching enabled. This mode is used for long sequences of short reads to consecutive addresses (i.e. shadowing)<br>• 11 - Illegal. Caching must be enabled when Prefetching is enabled. This eliminates the need for a complex prefetch-flushing mechanism. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW/1L | **Lock Enable (LE):** When set, setting the WP bit will cause SMI.<br>When cleared, setting the WP bit will not cause SMI. Once set, this bit can only be cleared by a PLTRST#.<br>When this bit is set, EISS - bit [5] of this register is locked down. |
| 0 | 0x0<br>RW | **Write Protect Disable (WPD):** When set, access to the BIOS space is enabled for both read and write cycles to BIOS.<br>When cleared, only read cycles are permitted to the FWH or SPI flash.<br>When this bit is written from a 0 to a 1 and the LE bit is also set, an SMI# is generated.<br>This ensures that only SMM code can update BIOS. |

## 56.2.13 PCI Clock Control (PCCTL)—Offset E0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:0] + E0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 9 | 0x0<br>RO/V | **CLKRUN# Buffer Enable Override (CLKRUN_EN_OVR):** When set to '1', SW is in control of the CLKRUN# buffer enable and the value in CLKRUN_EN_VAL will be propagated to the output buffer enable.<br>When this bit is '0', HW will determine the value of the buffer enable. |
| 8 | 0x0<br>RO/V | **CLKRUN# Override (CLKRUN_OVR):** When set to '1', SW is in control of the CLKRUN# pin and the value in CLKRUN_VAL will be propagated to the output pin.<br>When this bit is '0', HW will determine the value of the pin. |
| 7 | 0x0<br>RO/V | **CLKRUN# Buffer Enable Value (CLKRUN_EN_VAL):** Either HW or SW may own control of the CLKRUN# pin.<br>This bit provides the value to drive on the active low CLKRUN# buffer enable if CLKRUN_EN_OVR is set to '1'. |
| 6 | 0x0<br>RO/V | **CLKRUN# Pin Output Value (CLKRUN_VAL):** Either HW or SW may own control of the CLKRUN# pin.<br>This bit provides the value to drive on the pin if CLKRUN_OVR is set to '1'. |
| 5 | 0x0<br>RO/V | **Stop PCI# Value (STP_PCI_VAL):** Either Hardware or Software may own control of the internal STP_PCI#.<br>This bit provides the value to drive on the STP_PCI# if STP_PCI_OVR is set to 1.<br>***Note:***<br>1. SW cannot control the STP_PCI# pin while PLTRST# is asserted (the pin will be at its reset default value). |
| 4 | 0x0<br>RO/V | **Stop PCI# Override (STP_PCI_OVR):** When set to 1, Firmware is in control of the STP_PCI# and the value in STP_PCI_VAL will be propagated to the internal STP_PCI#.<br>When this bit is '0', HW will determine the value of the pin.<br>***Note:***<br>1. BIOS cannot control the STP_PCI# pin while PLTRST# is asserted (the pin will be at its reset default value). |
| 3:2 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved |
| 1 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved. |
| 0 | 0x0<br>RW | **Clock Run Enable (CLKRUN_EN):** Enables the CLKRUN# logic to stop the PCI clocks.<br>If the SLP_EN bit is set, then the SoC will drive CLKRUN# low.<br>This will keep the PCI and LPC clocks running on the way to the sleeping state.<br>This is required to meet an LPC specification.<br>This does not necessarily mean that the CLKRUN_EN bit is forced low when SLP_EN is set.<br>Even though the CLKRUN# signal will be low when SLP_EN is set, the state of the CLKRUN_EN bit is ignored when SLP_EN bit is set.<br>This gives flexibility in the implementation. |

§ §

# 57 Primary to Side Band Bridge - B0, D31, F1

## 57.1 Introduction and Index

The host-accessible registers for the Primary to Side Band (P2SB) Bridge are described here.

The legend for the register access codes are described in Chapter 43, “Introduction.”

### 57.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 1. The offset addresses are listed.

**Table 57-1. Summary of PCI Configuration Registers—0/31/1**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19DD8086 | "PCI Identifier (PCIID)—Offset 0h" on page 2663 |
| 4 | 2 | 0004 | "PCI Command (PCICMD)—Offset 4h" on page 2664 |
| 6 | 2 | 0000 | "PCI Status (PCISTS)—Offset 6h" on page 2665 |
| 8 | 1 | 00 | "Revision ID (PCIRID)—Offset 8h" on page 2665 |
| 9 | 4 | 00058000 | "Class Code (PCICC)—Offset 9h" on page 2666 |
| D | 1 | 00 | "PCI Master Latency Timer (PCIMLT)—Offset Dh" on page 2666 |
| E | 2 | 00 | "PCI Header Type (PCIHTYPE)—Offset Eh" on page 2666 |
| 10 | 4 | 00000004 | "Sideband Register Access BAR (SBREG_BAR)—Offset 10h" on page 2667 |
| 14 | 4 | 00000000 | "Sideband Register BAR High DWORD (SBREG_BARH)—Offset 14h" on page 2667 |
| 2C | 4 | 00000000 | "PCI Subsystem Identifiers (PCIHSS)—Offset 2Ch" on page 2668 |
| 50 | 2 | 00F8 | "VLW Bus:Device:Function (VBDF)—Offset 50h" on page 2668 |
| 52 | 2 | 00F8 | "ERROR Bus:Device:Function (EBDF)—Offset 52h" on page 2669 |
| 54 | 4 | 0000C700 | "Routing Configuration (RCFG)—Offset 54h" on page 2670 |
| 60 | 1 | 00 | "High Performance Event Timer Configuration (HPTC)—Offset 60h" on page 2670 |
| 64 | 2 | 0000 | "IOxAPIC Configuration (IOAC)—Offset 64h" on page 2671 |
| 6C | 2 | 00F8 | "IOxAPIC Bus:Device:Function (IBDF)—Offset 6Ch" on page 2671 |
| 70 | 2 | 00F8 | "HPET Bus:Device:Function (HBDF)—Offset 70h" on page 2672 |
| 80 | 4 | 00000000 | "Sideband Register posted 0 (SBREGPOSTED0)—Offset 80h" on page 2672 |
| 84 | 4 | 00000000 | "Sideband Register posted 1 (SBREGPOSTED1)—Offset 84h" on page 2672 |
| 88 | 4 | 00000000 | "Sideband Register posted 2 (SBREGPOSTED2)—Offset 88h" on page 2673 |
| 8C | 4 | 00000000 | "Sideband Register posted 3 (SBREGPOSTED3)—Offset 8Ch" on page 2673 |
| 90 | 4 | 00000000 | "Sideband Register posted 4 (SBREGPOSTED4)—Offset 90h" on page 2673 |
| 94 | 4 | 00000000 | "Sideband Register posted 5 (SBREGPOSTED5)—Offset 94h" on page 2673 |
| 98 | 4 | 00000000 | "Sideband Register posted 6 (SBREGPOSTED6)—Offset 98h" on page 2674 |
| 9C | 4 | 00000000 | "Sideband Register posted 7 (SBREGPOSTED7)—Offset 9Ch" on page 2674 |
| A0 | 4 | 00060010 | "Display Bus:Device:Function (DISPBDF)—Offset A0h" on page 2674 |
| A4 | 2 | 0000 | "ICC Register Offsets (ICCOS)—Offset A4h" on page 2675 |
| B0 | 4 | 00000000 | "Endpoint Mask 0 (EPMASK0)—Offset B0h" on page 2675 |
| B4 | 4 | 00000000 | "Endpoint Mask 1 (EPMASK1)—Offset B4h" on page 2675 |
| B8 | 4 | 00000000 | "Endpoint Mask 2 (EPMASK2)—Offset B8h" on page 2676 |
| BC | 4 | 00000000 | "Endpoint Mask 3 (EPMASK3)—Offset BCh" on page 2676 |
| C0 | 4 | 00000000 | "Endpoint Mask 4 (EPMASK4)—Offset C0h" on page 2676 |
| C4 | 4 | 00000000 | "Endpoint Mask 5 (EPMASK5)—Offset C4h" on page 2676 |
| C8 | 4 | 00000000 | "Endpoint Mask 6 (EPMASK6)—Offset C8h" on page 2677 |
| CC | 4 | 00000000 | "Endpoint Mask 7 (EPMASK7)—Offset CCh" on page 2677 |
| D0 | 4 | 00000000 | "SBI Address (SBIADDR)—Offset D0h" on page 2677 |

## Table 57-1. Summary of PCI Configuration Registers—0/31/1 (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| D4 | 4 | 00000000 | "SBI Data (SBIDATA)—Offset D4h" on page 2678 |
| D8 | 2 | 0000 | "SBI Status (SBISTAT)—Offset D8h" on page 2678 |
| DA | 2 | 0000 | "SBI Routing Identification (SBIRID)—Offset DAh" on page 2679 |
| DC | 4 | 00000000 | "SBI Extended Address (SBIEXTADDR)—Offset DCh" on page 2679 |
| E0 | 4 | 00000007 | "P2SB Control (P2SBC)—Offset E0h" on page 2680 |
| F0 | 4 | 00000000 | "Unsupported Request Error Status (URES)—Offset F0h" on page 2681 |
| F4 | 4 | 00000000 | "Unsupported Request Error Control (UREC)—Offset F4h" on page 2681 |
| F8 | 4 | 00000F00 | "Manufacturer's ID (MANID)—Offset F8h" on page 2682 |

## 57.2 Registers in Configuration Space

### 57.2.1 PCI Identifier (PCIID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 0h

**Default:** 19DD8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19dd<br>RO/V | **Device Identification (DID):** The upper 8-bits of this field can be overridden by the SoC.<br><br>**Power Well:** Core |
| 15:0 | 0x8086<br>RO | **Vendor Identification (VID):** Indicates Intel<br><br>**Power Well:** Core |

## 57.2.2 PCI Command (PCICMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 4h

**Default:** 0004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 10 | 0x0<br>RO | **Interrupt Disable (INTD):** P2SB does not issue any interrupts on its own behalf<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FB2BE):** Not applicable<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **SERR# Enable (SEE):** This bit enables parity error reporting to the SoC Interrupt Error Handler (IEH)<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Parity Error Response Enable (PEE):** This bit controls the device's response to parity error.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGA):** Not applicable.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Memory Write & Invalidate Enable (MWIE):** Not applicable.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Not applicable.<br><br>**Power Well:** Core |
| 2 | 0x1<br>RO | **Bus Master Enable (BME):** Bus mastering cannot be disabled as this device acts as a proxy for non-PCI devices.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** This bit controls the P2SB acceptance of PCI MMIO BARs only. Other legacy regions are unaffected by this bit.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **I/O Space Enable (IOSE):** Legacy regions are unaffected by this bit.<br><br>**Power Well:** Core |

## 57.2.3 PCI Status (PCISTS)—Offset 6h

The P2SB does not issue any interrupts, signal any errors, or provide any additional capability structures. This register has no functionality.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 6h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **Detected Parity Error (DPE):** This register bit is set when parity error is detected.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW/1C | **Signaled System Error (SSE):** This register sets when PCICMD.SE bit and PCISTS.DPE bit is set<br><br>**Power Well:** Core |
| 13:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |

## 57.2.4 Revision ID (PCIRID)—Offset 8h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:1] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates the part revision. This field resets to 0 and can be overridden by the SoC circuitry during the SoC power-up sequence<br><br>**Power Well:** Core |

## 57.2.5 Class Code (PCICC)—Offset 9h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 9h

**Default:** 00058000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **RSVD0:** Reserved |
| 23:16 | 0x5<br>RO | **Base Class Code (BCC):** Indicates a Memory Controller device class.<br><br>**Power Well:** Core |
| 15:8 | 0x80<br>RO | **Sub-Class Code (SCC):** Indicates an unspecified sub-class other Memory Controller.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RO | **Programming Interface (PI):** No programming interface.<br><br>**Power Well:** Core |

## 57.2.6 PCI Master Latency Timer (PCIMLT)—Offset Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:1] + Dh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |

## 57.2.7 PCI Header Type (PCIHTYPE)—Offset Eh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:1] + Eh

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Multi-Function Device (MFD):** Indicates that this is part of a multi-function device.<br><br>**Power Well:** Core |
| 6:0 | 0x0<br>RO | **Header Type (HTYPE):** Indicates a generic Type 0 PCI device header.<br><br>**Power Well:** Core |

## 57.2.8 Sideband Register Access BAR (SBREG_BAR)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **Register Base Address (RBA):** Lower DWORD of the base address for the sideband register access BAR.<br><br>**Power Well:** Core |
| 23:4 | 0x0<br>RO | **Hardwired to 0 to request a BAR of 16MB (HW202RB16MB):** Hardwired to 0 to request a BAR of 16MB<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Prefetchable (PREF):** Indicates this is not prefetchable.<br><br>**Power Well:** Core |
| 2:1 | 0x2<br>RO | **Address Type (ATYPE):** Indicates that the BAR can be defined to be located anywhere in 64-bit Host Root Memory Space.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Space Type (STYPE):** Indicates Memory Space<br><br>**Power Well:** Core |

## 57.2.9 Sideband Register BAR High DWORD (SBREG_BARH)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Register Base Address (RBAH):** Upper DWORD of the base address for the sideband register access BAR.<br><br>**Power Well:** Core |

## 57.2.10 PCI Subsystem Identifiers (PCIHSS)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Written by BIOS. Not used by hardware.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Written by BIOS. Not used by hardware.<br><br>**Power Well:** Core |

## 57.2.11 VLW Bus:Device:Function (VBDF)—Offset 50h

This register specifies the bus:device:function ID that the VLW VDM will use for its Requester ID. This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the VLW message.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 50h

**Default:** 00F8h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Bus Number (BUS):** Reserved.<br><br>**Power Well:** Core |
| 7:3 | 0x1f<br>RW | **Device Number (DEV):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Function Number (FUNC):** Reserved.<br><br>**Power Well:** Core |

## 57.2.12 ERROR Bus:Device:Function (EBDF)—Offset 52h

This register specifies the bus:device:function ID that the Error Signaling messages VDM will use for its Requester ID. This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the VLW message.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 52h

**Default:** 00F8h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Bus Number (BUS):** Reserved.<br><br>**Power Well:** Core |
| 7:3 | 0x1f<br>RW | **Device Number (DEV):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Function Number (FUNC):** Reserved.<br><br>**Power Well:** Core |

## 57.2.13 Routing Configuration (RCFG)—Offset 54h

This register contains information used for routing transactions between primary and sideband interfaces.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 54h

**Default:** 0000C700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:8 | 0xc7<br>RW | **Reserved Page Register Destination ID (RPRID):** Specifies the IOSF-SB destination ID for sending Reserved Page Register cycles (e.g. Port 80h). By default this will load to the ID of the LPC device depending on which has been strapped active in the system.<br><br>**Power Well:** Core |
| 7:1 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **RTC Shadow Enable (RSE):** When set, all IO writes to the RTC will be also sent to the PMC. This allows cases where the battery-backed storage is in an external PMIC device.<br><br>**Power Well:** Core |

## 57.2.14 High Performance Event Timer Configuration (HPTC)—Offset 60h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:1] + 60h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Address Enable (AE):** When set, the P2SB will decode the High Performance Event Timer (HPET) Host Root memory-address range selected by bits 1:0 below.<br><br>**Power Well:** Core |
| 6:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RW | **Address Select (AS):** This 2-bit field selects 1 of 4 possible memory address ranges for the High-Performance Event Timer (HPET) functionality. The encodings are:<br>• 00: FED0_0000h - FED0_03FFFh<br>• 01: FED0_1000h - FED0_13FFFh<br>• 10: FED0_2000h - FED0_23FFFh<br>• 11: FED0_3000h - FED0_33FFFh<br><br>**Power Well:** Core |

## 57.2.15 IOxAPIC Configuration (IOAC)—Offset 64h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 64h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Address Enable (AE):** When set, the P2SB will decode the IOxAPIC Host Root memory-address range selected by bits 7:0 below.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **APIC Range Select (ASEL):** These eight bits define the eight address bits 19:12 for the IOxAPIC range. The default value of 00h enables compatibility with prior Intel products as an initial value. This value must not be changed by software unless the IOxAPIC Address Enable (AE) bit is cleared.<br><br>**Power Well:** Core |

## 57.2.16 IOxAPIC Bus:Device:Function (IBDF)—Offset 6Ch

This register specifies the PCI Bus:Device:Function ID that the IOxAPIC will use in the following:

- As the Requester ID when initiating Interrupt Messages to the CPU.
- As the Completer ID when responding to the reads targeting the IOxAPIC Memory-Mapped I/O (MMIO) registers.

This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if a unique Bus:Device:Function number is required for the internal IOxAPIC.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 6Ch

**Default:** 00F8h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Bus Number (BUS):** Reserved.<br><br>**Power Well:** Core |
| 7:3 | 0x1f<br>RW | **Device Number (DEV):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Function Number (FUNC):** Reserved.<br><br>**Power Well:** Core |

## 57.2.17 HPET Bus:Device:Function (HBDF)—Offset 70h

This register specifies the PCI Bus:Device:Function ID that the integrated High-Performance Event Timer (HPET) device will use in the following:

- As the Requester ID when initiating Interrupt Messages to the CPU.
- As the Completer ID when responding to the reads targeting the corresponding HPET Memory-Mapped I/O (MMIO) registers.

This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique Bus:Device:Function number is required for the corresponding HPET.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 70h

**Default:** 00F8h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Bus Number (BUS):** Reserved.<br><br>**Power Well:** Core |
| 7:3 | 0x1f<br>RW | **Device Number (DEV):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Function Number (FUNC):** Reserved.<br><br>**Power Well:** Core |

## 57.2.18 Sideband Register posted 0 (SBREGPOSTED0)—Offset 80h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 80h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 0 (SBREGPOSTED0):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 31-0.<br><br>**Power Well:** Core |

## 57.2.19 Sideband Register posted 1 (SBREGPOSTED1)—Offset 84h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 84h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 1 (SBREGPOSTED1):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 63-32.<br><br>**Power Well:** Core |

## 57.2.20 Sideband Register posted 2 (SBREGPOSTED2)—Offset 88h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 88h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 0 (SBREGPOSTED2):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 95-64.<br><br>**Power Well:** Core |

## 57.2.21 Sideband Register posted 3 (SBREGPOSTED3)—Offset 8Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 8Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 3 (SBREGPOSTED3):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 127-96.<br><br>**Power Well:** Core |

## 57.2.22 Sideband Register posted 4 (SBREGPOSTED4)—Offset 90h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 4 (SBREGPOSTED4):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 159-128.<br><br>**Power Well:** Core |

## 57.2.23 Sideband Register posted 5 (SBREGPOSTED5)—Offset 94h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 5 (SBREGPOSTED5):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 191-160.<br><br>**Power Well:** Core |

## 57.2.24 Sideband Register posted 6 (SBREGPOSTED6)—Offset 98h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 6 (SBREGPOSTED6):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 223-192.<br><br>**Power Well:** Core |

## 57.2.25 Sideband Register posted 7 (SBREGPOSTED7)—Offset 9Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Sideband Register posted 7 (SBREGPOSTED7):** One hot masks for setting SBREG to posted for IOSF-SB endpoint IDs 255-224.<br><br>**Power Well:** Core |

## 57.2.26 Display Bus:Device:Function (DISPBDF)—Offset A0h

This register specifies the PCI Bus:Device:Function ID that the Display initiated upstream RAVDMs will use for its Requester ID. This will also be used for the claiming these Route-by-ID RAVDMs downstream.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + A0h

**Default:** 00060010h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 18:16 | 0x6<br>RW | **Display Target Block (DTBLK):** This register contains the Target BLK field that will be used when sending RAVDM messages to the CPU Complex North Display.<br><br>**Power Well:** Core |
| 15:8 | 0x0<br>RW | **Bus Number (BUS):** The bus number of the Display in the CPU Complex.<br><br>**Power Well:** Core |
| 7:3 | 0x2<br>RW | **Device Number (DEV):** The bus number of the Display in the CPU Complex.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Function Number (FUNC):** The function number of the Display in the CPU Complex<br><br>**Power Well:** Core |

## 57.2.27 ICC Register Offsets (ICCOS)—Offset A4h

This register contains the offsets to be used when sending RAVDMs to the Integrated Clock Controller. Each of the two spaces decoded for the ICC have a separate base address that will be used when sending those transactions on IOSF-SB to the ICC.

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + A4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Modulator Control Address Offset (MODBASE):** This specifies the upper 8b for the 16b address that will be used for sending RAVDM access that target the Modulator Control range of the ICC (FFF00h - FFFFFh).<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Buffer Address Offset (BUFBASE):** This specifies the upper 8b for the 16b address that will be used for sending RAVDM access that target the Buffer range of the ICC (FFE00h - FFEFFh).<br><br>**Power Well:** Core |

## 57.2.28 Endpoint Mask 0 (EPMASK0)—Offset B0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V/L | **Endpoint Mask 0 (EPMASK0):** One hot masks for disabling IOSF-SB endpoint IDs 31-0.<br><br>**Power Well:** Core |

## 57.2.29 Endpoint Mask 1 (EPMASK1)—Offset B4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + B4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V/L | **Endpoint Mask 1 (EPMASK1):** One hot masks for disabling IOSF-SB endpoint IDs 63-32.<br><br>**Power Well:** Core |

## 57.2.30 Endpoint Mask 2 (EPMASK2)—Offset B8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + B8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V/L | **Endpoint Mask 2 (EPMASK2):** One hot masks for disabling IOSF-SB endpoint IDs 95-64.<br>**Power Well:** Core |

## 57.2.31 Endpoint Mask 3 (EPMASK3)—Offset BCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + BCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V/L | **Endpoint Mask 3 (EPMASK3):** One hot masks for disabling IOSF-SB endpoint IDs 127-96.<br>**Power Well:** Core |

## 57.2.32 Endpoint Mask 4 (EPMASK4)—Offset C0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V/L | **Endpoint Mask 4 (EPMASK4):** One hot masks for disabling IOSF-SB endpoint IDs 128-159.<br>**Power Well:** Core |

## 57.2.33 Endpoint Mask 5 (EPMASK5)—Offset C4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0 RW/V/L | **Endpoint Mask 5 (EPMASK5):** One hot masks for disabling IOSF-SB endpoint IDs 191-160.<br>**Power Well:** Core |

## 57.2.34 Endpoint Mask 6 (EPMASK6)—Offset C8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V/L | **Endpoint Mask 6 (EPMASK6):** One hot masks for disabling IOSF-SB endpoint IDs 223-192.<br><br>**Power Well:** Core |

## 57.2.35 Endpoint Mask 7 (EPMASK7)—Offset CCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V/L | **Endpoint Mask 7 (EPMASK7):** One hot masks for disabling IOSF-SB endpoint IDs 255-224.<br><br>**Power Well:** Core |

## 57.2.36 SBI Address (SBIADDR)—Offset D0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **Destination Port ID (DESTID):** The content of this register field is sent in the IOSF Sideband Message Register Access dest field.<br><br>**Power Well:** Core |
| 23:20 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 19:16 | 0x0<br>RW | **Root Space (RS):** Destination IOSF-SB Root Space.<br>***Note:*** This register may only be written during manufacturing test. P2SB will only accept writes to this register from transactions with a SAI equal to the SBI_RS_ACCESS_SAI parameter. This should be assigned to the SAI used by the functional test module (typically TAM) that will perform this register write on IOSF-P.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW | **Address Offset (OFFSET):** Register address offset.<br>The content of this register field is sent in the IOSF Sideband Message Register Access address(15:0) field.<br><br>**Power Well:** Core |

## 57.2.37 SBI Data (SBIDATA)—Offset D4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + D4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Data (DATA):** The content of this register field is sent on the IOSF sideband Message Register Access data(31:0) field.<br><br>**Power Well:** Core |

## 57.2.38 SBI Status (SBISTAT)—Offset D8h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + D8h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:8 | 0x0<br>RW | **Opcode (OPCODE):** This is the Opcode sent in the IOSF sideband message.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **Posted (POSTED):** When set to 1, the message will be send as a posted message instead of non-posted.<br>This should only be used if the receiver is known to support posted operations for the specified operation.<br><br>**Power Well:** Core |
| 6:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 2:1 | 0x0<br>RW/V | **Response Status (RESPONSE):**<br>• 00 - Successful<br>• 01 - Unsuccessful / Not Supported<br>• 10 - Powered Down<br>• 11 - Multi-cast Mixed<br>This register reflects the response status for the previously completed transaction.<br>The value of this register is only meaningful if SBISTAT.INITRDY is zero.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/1S | **Initiate/ Ready# (INITRDY):**<br>• 0: The IOSF sideband interface is ready for a new transaction<br>• 1: The IOSF sideband interface is busy with the previous transaction.<br>A write to set this register bit to 1 will trigger an IOSF sideband message on the private IOSF sideband interface.<br>The message will be formed based on the values programmed in the Sideband Message Interface Register Access registers.<br>Software needs to ensure that the interface is not busy (SBISTAT.INITRDY is clear) before writing to this register.<br><br>**Power Well:** Core |

## 57.2.39 SBI Routing Identification (SBIRID)—Offset DAh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + DAh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:12 | 0x0<br>RW | **First Byte Enable (FBE):** The content of this field is sent in the IOSF Sideband Register Access FBE field.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 10:8 | 0x0<br>RW | **Base Address Register (BAR):** The contents of this field are sent in the IOSF Sideband Register Access BAR field.<br>This should be zero performing a Memory Mapped operation to a PCI compliant device.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Function ID (FID):** The contents of this field are sent in the IOSF Sideband Register access FID field. This field should generally remain at zero unless specifically required by a particular application.<br><br>**Power Well:** Core |

## 57.2.40 SBI Extended Address (SBIEXTADDR)—Offset DCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + DCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Extended Address (ADDR):** The content of this register field is sent on the IOSF sideband Message Register Access address(48:32) field.<br>This must be set to all 0 if 16b addressing is desired.<br><br>**Power Well:** Core |

## 57.2.41 P2SB Control (P2SBC)—Offset E0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + E0h

**Default:** 00000007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RW | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 26 | 0x0<br>RW | **Data Parity Error Enable (DPEE):** IOSFP data parity error handling (EP bit) enabling bit<br><br>**Power Well:** Core |
| 25 | 0x0<br>RW | **Command Parity Error Enable (CPEE):** IOSFP command parity error handling enabling bit<br><br>**Power Well:** Core |
| 24 | 0x0<br>RW | **Data Phase Parity Error Enable (DPPEE):** IOSFP data phase parity error handling enabling bit<br><br>**Power Well:** Core |
| 23:18 | 0x0<br>RW | **Reserved (RSVD_3):** Reserved.<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW/O | **Endpoint Mask Lock (MASKLOCK):** Locks the value of the eight Endpoint Mask (EPMASK[0-7]) registers.<br>Once this value is written to a one it may only be cleared by a reset.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **PGCB Clock Gating Enable (PGCBCGE):** When asserted, the P2SB can de-assert the clock request to disable the PGCB clock dynamically when it reaches the power-down idle state.<br><br>**Power Well:** Core |
| 15:9 | 0x0<br>RW | **Reserved: (RSVD_1):** Hardware will not allocate these bits for any purpose in the future so that BIOS can always write to the HIDE field with this byte enabled without having to remember any other field values since it will not be possible to do a read-modify-write when the device is hidden.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Hide Device (HIDE):** When this bit is set, the P2SB will return 1s on any PCI Configuration Read on IOSF-P. All other transactions including PCI Configuration Writes are unaffected by this.<br>This does not affect reads performed on the IOSF-SB interface.<br><br>**Power Well:** Core |
| 7:3 | 0x0<br>RW | **Reserved (RSVD_2):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x7<br>RW | **Max Writes Pending (MAXW):** This controls the maximum number of outstanding writes on IOSF-SB initiated by MMIO writes to the SBREG_BAR.<br>Once the number of SBREG_BAR writes that are issued, but not completed, on IOSF-SB is equal to this value, no new requests will be forwarded until the completions are received.<br>A value of zero will have the same behavior as the value of one (single write outstanding).<br><br>**Power Well:** Core |

## 57.2.42 Unsupported Request Error Status (URES)—Offset F0h

This register is only reset by a loss of core power

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + F0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW/1C | **Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unsupported Request on IOSF Primary interface that is not considered Advisory Non-Fatal<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |

## 57.2.43 Unsupported Request Error Control (UREC)—Offset F4h

This register is only reset by a loss of core power.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + F4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Unsupported Request Reporting Enable (URRE):** If set to 1 by software, it allows reporting of an Unsupported Request as a system error.<br><br>**Power Well:** SUS |
| 7:0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |

## 57.2.44 Manufacturer's ID (MANID)—Offset F8h

This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetID message.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:1] + F8h

**Default:** 00000F00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 27:24 | 0x0<br>RO/V | **Dot portion of Process ID (DOT):** Indicates the dot process<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO/V | **Manufacturing Stepping Identifier (MSID):** This field is incremented for each stepping of the part.<br>***Note:*** This field can be used by software to differentiate steppings when the Stepping Revision ID may not change.<br><br>**Power Well:** Core |
| 15:8 | 0xf<br>RO | **Manufacturing Identifier (MID):** 0Fh = Intel<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RO/V | **Process portion of process ID (PROC):** Indicates the process.<br>The dot portion of the process is reflected in bits (27:24)<br><br>**Power Well:** Core |

§ §

# 58 Power Management Controller - B0, D31, F2

## 58.1 Introduction and Index

The host-accessible registers for the Power Management Controller (PMC) are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 58.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 2. The offset addresses are listed.

**Table 58-1. Summary of PCI Configuration Registers—0/31/2**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19DE8086 | "PCI Identifier (PCIID)—Offset 0h" on page 2689 |
| 40 | 4 | 00000001 | "ACPI Base Address (ABASE)—Offset 40h" on page 2689 |
| 44 | 4 | 00000000 | "ACPI Control (ACTL)—Offset 44h" on page 2690 |
| A0 | 4 | A0000000 | "General PM Configuration A (GEN_PMCON_A)—Offset A0h" on page 2691 |
| A4 | 4 | 00004007 | "General PM Configuration B (GEN_PMCON_B)—Offset A4h" on page 2693 |
| AC | 4 | 00000000 | "Extended Test Mode Register 3 (ETR3)—Offset ACh" on page 2695 |

## 58.1.2 Host Memory Space—PWRMBASE

### Table 58-2. Summary of Memory Mapped I/O Registers—PWRMBASE

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | FFFFFFFF | "Wake Alarm Device Timer: AC (WADT_AC)—Offset 0h" on page 2696 |
| 4 | 4 | FFFFFFFF | "Wake Alarm Device Timer: DC (WADT_DC)—Offset 4h" on page 2697 |
| 8 | 4 | FFFFFFFF | "Wake Alarm Device Expired Timer: AC (WADT_EXP_AC)—Offset 8h" on page 2698 |
| C | 4 | FFFFFFFF | "Wake Alarm Device Expired Timer: DC (WADT_EXP_DC)—Offset Ch" on page 2698 |
| 10 | 4 | 05000000 | "Power and Reset Status (PRSTS)—Offset 10h" on page 2699 |
| 18 | 4 | 00000020 | "Power Management Configuration Reg 1 (PM_CFG)—Offset 18h" on page 2700 |
| 1C | 4 | 00000000 | "PCH Power Management Status (PCH_PM_STS)—Offset 1Ch" on page 2704 |
| 24 | 4 | 00000000 | "PCH Power Management Status (PCH_PM_STS2)—Offset 24h" on page 2705 |
| F0 | 4 | 200F0DE4 | "ADR Enable (ADR_EN)—Offset F0h" on page 2707 |
| F8 | 4 | 00000000 | "System Time Capture Trigger (SYSTIMCT)—Offset F8h" on page 2709 |
| FC | 4 | 00000000 | "ACPI Timer Control (ACPI_TMR_CTL)—Offset FCh" on page 2709 |
| 120 | 4 | 00000432 | "GPIO Configuration (GPIO_CFG)—Offset 120h" on page 2710 |
| 124 | 4 | 00000000 | "Global Reset Causes (GBLRST_CAUSE0)—Offset 124h" on page 2711 |
| 128 | 4 | 00000000 | "Global Reset Causes (GBLRST_CAUSE1)—Offset 128h" on page 2713 |
| 12C | 4 | 00000000 | "Host Partition Reset Causes (HPR_CAUSE0)—Offset 12Ch" on page 2714 |
| 200 | 4 | 0000FFFF | "modPHY Power Management Configuration Reg 1 (MODPHY_PM_CFG1)—Offset 200h" on page 2715 |
| 204 | 4 | 05000000 | "modPHY Power Management Configuration Reg 2 (MODPHY_PM_CFG2)—Offset 204h" on page 2716 |
| 208 | 4 | 00000000 | "modPHY Power Management Configuration Reg 3 (MODPHY_PM_CFG3)—Offset 208h" on page 2718 |
| 20C | 4 | 00000000 | "modPHY Power Management Configuration Reg 4 (MODPHY_PM_CFG4)—Offset 20Ch" on page 2719 |
| 3F0 | 4 | 00000000 | "IE_STS (IE_STS)—Offset 3F0h" on page 2720 |
| 3F4 | 4 | 00000000 | "ME_STS (ME_STS)—Offset 3F4h" on page 2722 |
| 620 | 4 | 00000000 | "Static PG Related Function Disable Register 1 (ST_PG_FDIS_PMC_1)—Offset 620h" on page 2723 |
| 628 | 4 | 00000000 | "Non-Static PG Related Function Disable Register 1 (NST_PG_FDIS_1)—Offset 628h" on page 2724 |
| 640 | 4 | 00000000 | "Non-Static PG Fuse Disable Read 1 Register (N_STPG_FUSE_SS_DIS_RD_1)—Offset 640h" on page 2724 |
| 644 | 4 | 00000000 | "Static PG Fuse and Soft Strap Disable Read Register 2 (STPG_FUSE_SS_DIS_RD_2)—Offset 644h" on page 2725 |

### 58.1.3 Host I/O Space—ABASE

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 58-3. Summary of I/O Registers—ABASE**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "Power Management 1 Enables and Status (PM1_EN_STS)—Offset 0h" on page 2726 |
| 4 | 4 | 00000000 | "Power Management 1 Control —Offset 4h" on page 2728 |
| 8 | 4 | 00000000 | "Power Management 1 Timer (PM1_TMR)—Offset 8h" on page 2729 |
| 30 | 4 | 00000002 | "SMI Control and Enable (SMI_EN)—Offset 30h" on page 2730 |
| 34 | 4 | 00000000 | "SMI Status Register (SMI_STS)—Offset 34h" on page 2732 |
| 40 | 4 | 00000000 | "General Purpose Event Control (GPE_CTRL)—Offset 40h" on page 2735 |
| 44 | 4 | 00000000 | "Device Trap Status Register (DEVTRAP_STS)—Offset 44h" on page 2736 |
| 50 | 4 | 00000000 | "PM2a Control Block (PM2A_CNT_BLK)—Offset 50h" on page 2737 |
| 54 | 4 | 00002000 | "Over-Clocking WDT Control (OC_WDT_CTL)—Offset 54h" on page 2737 |
| 80 | 4 | 00000000 | "General Purpose Event 0 Status [31:0] (GPE0_STS_31_0)—Offset 80h" on page 2738 |
| 84 | 4 | 00000000 | "General Purpose Event 0 Status [63:32] (GPE0_STS_63_32)—Offset 84h" on page 2739 |
| 88 | 4 | 00000000 | "General Purpose Event 0 Status [95:64] (GPE0_STS_95_64)—Offset 88h" on page 2739 |
| 8C | 4 | 00000000 | "General Purpose Event 0 Status [127:96] (GPE0_STS_127_96)—Offset 8Ch" on page 2740 |
| 90 | 4 | 00000000 | "General Purpose Event 0 Enable [31:0] (GPE0_EN_31_0)—Offset 90h" on page 2742 |
| 94 | 4 | 00000000 | "General Purpose Event 0 Enable [63:32] (GPE0_EN_63_32)—Offset 94h" on page 2742 |
| 98 | 4 | 00000000 | "General Purpose Event 0 Enable [95:64] (GPE0_EN_95_64)—Offset 98h" on page 2742 |
| 9C | 4 | 00000000 | "General Purpose Event 0 Enable [127:96] (GPE0_EN_127_96)—Offset 9Ch" on page 2743 |

### 58.1.4 Host I/O Space—Fixed Address

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 58-4. Summary of I/O Registers—Fixed Addresses**

| Address (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| CF9 | 1 | 0 | "Reset Control Register (RST_CNT)" on page 2745 |

### 58.1.5 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 58-5. Summary of Sideband Registers—0x82**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| B0 | 00000000 | "Advanced Power Management Status and Control Port (APM_STS_CNT)—Offset B0h" on page 2746 |
| B4 | 00000000 | "Misc SB Reg0 (MISC_SBR0)—Offset B4h" on page 2746 |

## 58.2 Registers in Configuration Space

### 58.2.1 PCI Identifier (PCIID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + 0h

**Default:** 19DE8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19DE<br>RO/V | **Device Identification (DID):** These bits are assigned by Intel. |
| 15:0 | 0x8086<br>RO | **Vendor Identification (VID):** Indicates Intel |

### 58.2.2 ACPI Base Address (ABASE)—Offset 40h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + 40h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW/L | **Base Address (BA):** Provides the 256 bytes of I/O space for ACPI logic.<br>This field is locked by GEN_PMCON_B.ACPI_BASE_LOCK. |
| 7:1 | 0x0<br>RO | **Reserved (RSVD_7_1):** Always 0. |
| 0 | 0x1<br>RO | **Space Type (STYPE):** Always 1 to indicate I/O space. |

## 58.2.3 ACPI Control (ACTL)—Offset 44h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_31_9):** Reserved. |
| 8 | 0x0<br>RW | **PWRM Enable (PWRM_EN):** When set, decode of the Host Memory Space range pointed by PWRMBASE is enabled. |
| 7 | 0x0<br>RW | **ACPI Enable (EN):** When set, decode of the I/O range pointed to by the ACPI base register is enabled and the ACPI power management function is enabled. |
| 6:3 | 0x0<br>RO | **Reserved (RSVD_6_3):** Reserved. |
| 2:0 | 0x0<br>RW | **SCI IRQ Select (SCIS):** Specifies on which IRQ the SCI will internally appear.<br>If not using the APIC, the SCI must be routed to IRQ[9-11], and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts.<br>If using the APIC, the SCI can also be mapped to IRQ20-23, and can be shared with other interrupts.<br>Bits - SCI Map<br>---- - -------<br>• 000 - IRQ9<br>• 001 - IRQ10<br>• 010 - IRQ11<br>• 011 - Reserved<br>• 100 - IRQ20 (only if APIC is enabled)<br>• 101 - IRQ21 (only if APIC is enabled)<br>• 110 - IRQ22 (only if APIC is enabled)<br>• 111 - IRQ23 (only if APIC is enabled)<br>When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC should be programmed for active-high reception.<br>When the interrupt is mapped to APIC interrupts 20 through 23, the APIC should be programmed for active-low reception. |

## 58.2.4 General PM Configuration A (GEN_PMCON_A)—Offset A0h

Usage: ACPI and Legacy.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + A0h

**Default:** A0000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW | **GBE Prep Non-fatal Timeout Enable (GBE_PREP_NFTO_EN):** When cleared, this bit causes the PMC to disable the GBE Prep non-fatal timeout, causing the PMC to wait indefinitely for the ACK STS (response) for Reset/Boot Prep message from the GBe MAC.<br>When set, a non-fatal timeout will apply to the Reset/Boot Prep handshake. |
| 30 | 0x0<br>RW | **RSVD:** Reserved |
| 29 | 0x1<br>RW | **RSVD:** Reserved |
| 28 | 0x0<br>RW | **After G3 PHY Power Enable (AG3_PP_EN):** This bit determines the Host software contribution to whether the LAN PHY is powered up after exiting G3.This bit is in the RTC power well. |
| 27 | 0x0<br>RW | **Sx PHY Power Enable (SX_PP_EN):** This bit determines the Host software contribution to whether the LAN PHY remains powered in an Sx/MOFF state that was entered from S0 (rather than from G3). |
| 26:24 | 0x0<br>RO | **Reserved (RSVD_26_24):** Reserved. |
| 23 | 0x0<br>RW | **DRAM Initialization Scratchpad Bit (DISB):** This bit does not affect hardware functionality in any way.<br>It is provided as a scratchpad bit that is maintained through main power well resets and CF9h-initiated resets.<br>If the bit is 1, then the DRAM initialization was interrupted. |
| 22 | 0x0<br>RO | **Reserved (RSVD_22):** Reserved. |
| 21 | 0x0<br>RO/V | **Memory Placed in Self-Refresh (MEM_SR):** This bit will be set to 1 if DRAM should have remained powered and held in Self-Refresh through the last power state transition (i.e. the last time the system left S0). The scenarios where this should be the case are:<br>• successful S3 entry and exit<br>• successful Host partition reset without power cycle<br>This indicates that memory was successfully placed into Self-Refresh.<br>This bit will be cleared whenever the SoC begins a transition out of S0. |
| 20:19 | 0x0<br>RO | **Reserved (RSVD_20_19):** Reserved. |
| 18 | 0x0<br>RW/1C/V | **Minimum SLP_S4# Assertion Width Violation Status (MS4V):** Hardware sets this bit when the SLP_S4# assertion width is less than the time programmed in the SLP_S4# Minimum Assertion Width field in the GEN_PMCON_B register. |
| 17 | 0x0<br>RO | **Reserved (RSVD_17):** Reserved. |
| 16 | 0x0<br>RW/1C/V | **Global Reset Status (GBL_RST_STS):** This bit is set after a global reset (not G3 or DeepSx) occurs. See the GEN_PMCON_B.HOST_RST_STS bit for potential usage models. |
| 15 | 0x0<br>RW | **Allow iCLK PLL Shutdown in C0 (ALLOW_ICLK_PLL_SD_INC0):** When this bit is left at '0' (default), PMC allows iCLK PLL Shutdown only when the CPU is in a non-C0 state. |
| 14 | 0x0<br>RW | **Dynamic mPHY CRI Clock Gate Override (MPHY_CRICLK_GATE_OVR):**<br>• 0: Enable CRI clock gating in the modPHY<br>• 1: Disable CRI clock gating in the modPHY<br>BIOS reading this register should always return the correct value. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 13 | 0x0<br>RW | **Allow OPI PLL Shutdown in C0 (ALLOW_OPI_PLL_SD_INC0):** When this bit is left at '0' (default), PMC allows OPI PLL Shutdown only when the CPU is in a non-C0 state. When set to '1', the OPI PLL can be shut down when the CPU is in C0 or Cx state assuming all other gating conditions are satisfied. |
| 12 | 0x0<br>RW | **Allow SPXB Clock Gating in C0 (ALLOW_SPXB_CG_INC0):**<br>When this bit is left at '0' (default), PMC allows SBXB Clock gating only when the CPU is in a non-C0 state. |
| 11 | 0x0<br>RO | **Reserved (RSVD_11):** Reserved. |
| 10 | 0x0<br>RW | **BIOS PCI Express Enable (BIOS_PCI_EXP_EN):** This bit acts as a global enable for the SCI associated with the PCI express ports.<br>If this bit is not set, then the various PCI Express ports cannot cause the PCI_EXP_STS bit to go active. |
| 9 | 0x0<br>RO/V | **Power Button Level (PWRBTN_LVL):** This read-only bit indicates the current state of the PMU_PWRBTN_N signal.<br>• 1= High<br>• 0 = Low |
| 8 | 0x0<br>RO | **Reserved (RSVD_8):** Reserved. |
| 7 | 0x0<br>RW | **Allow L1.LOW Entry During C0 (ALLOW_L1LOW_C0):**<br>When this bit is 0, the PMC only allows L1.LOW entry if the CPU is in Cx (EA = 0).<br>When set to 1, the PMC allows L1.LOW entry regardless of the value of EA. |
| 6 | 0x0<br>RW | **Allow L1.LOW Entry with OPI Voltage On (ALLOW_L1LOW_OPI_ON):**<br>When this bit is 0, the PMC only allows L1.LOW entry if the OPI voltage is off.<br>When this bit is 1, the PMC allows L1.LOW entry regardless of whether the OPI voltage is on/off. |
| 5 | 0x0<br>RW | **Allow L1.LOW Entry with CPU BCLK REQ Asserted (ALLOW_L1LOW_BCLKREQ_ON):**<br>When this bit is 0, the PMC only allows L1.LOW entry if the CPUs BCLK request is de-asserted.<br>When this bit is 1, the PMC allows L1.LOW entry regardless of whether the CPUs BCLK request is asserted/de-asserted. |
| 4 | 0x0<br>RW/L | **SMI Lock (SMI_LOCK):** When this bit is set, writes to the GLB_SMI_EN bit will have no effect.<br>Once the SMI_LOCK bit is set, writes of 0 to SMI_LOCK bit will have no effect (i.e. once set, this bit can only be cleared by host_prim_rst_b). |
| 3 | 0x0<br>RW/L | **Reserved:** Reserved. |
| 2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1:0 | 0x0<br>RW | **Period SMI Select (PER_SMI_SEL):** Software sets these bits to control the rate at which the periodic SMI# is generated:<br>• 00 = 64 seconds (default)<br>• 01 = 32 seconds<br>• 10 = 16 seconds<br>• 11 = 8 seconds<br>Tolerance for the timer is ± 1 second. |

## 58.2.5 General PM Configuration B (GEN_PMCON_B)—Offset A4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + A4h

**Default:** 00004007h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Reserved (RSVD_31_19):** Reserved. |
| 18 | 0x0<br>RW/L | **RSVD:** Reserved |
| 17 | 0x0<br>RW/L | **ACPI Base Lock (ACPI_BASE_LOCK):** When set to 1, this bit locks down the ACPI Base Address Register (ABASE) located at offset 40h.<br>The Base Address Field becomes read-only. This bit becomes locked when a value of 1b is written to it. |
| 16 | 0x0<br>RW/L | **RSVD:** Reserved |
| 15 | 0x0<br>RW | **PME B0 S5 Disable (PME_B0_S5_DIS):**<br>When cleared (default), wake events from PME_B0_STS are allowed in S5 if PME_B0_EN = 1.<br>When set to 1, this bit blocks wake events from PME_B0_STS in S5, regardless of the state of PME_B0_EN. The net effect of setting PME_B0_S5_DIS = 1 is described here:<br>• When PME_B0_EN = 0 and WOL Enable Override = 0: Don't Wake when in S5.<br>• When PME_B0_EN = 0 and WOL Enable Override = 1: Wake (LAN only) when in S5.<br>• When PME_B0_EN = 1 and WOL Enable Override = 0: Don't Wake when in S5.<br>• When PME_B0_EN = 1 and WOL Enable Override = 1: Wake (LAN only) when in S5.<br>This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 14 | 0x1<br>RW/1C | **SUS Well Power Failure (SUS_PWR_FLR):** This bit is set to '1' whenever SUS well power is lost, as indicated by pri_pwrgood_rst_b assertion. |
| 13 | 0x0<br>RW | **WOL Enable Override (WOL_EN_OVRD):**<br>When this bit is set to 1, the integrated LAN is enabled to wake the system from S5 regardless of the value in the PME_B0_EN bit in the GPE0_EN register. |
| 12 | 0x0<br>RW/L | **Disable SLP_X Stretching After SUS Well Power Up (DIS_SLP_X_STRCH_SUS_UP):**<br>When this bit is set to 1, all PMU_SLP_* pin stretching is disabled when powering up after a SUS well power loss.<br>When this bit is left at 0, PMU_SLP_* stretching will be performed after SUS power failure as enabled in various other fields. |
| 11:10 | 0x0<br>RW/L | **SLP_S3# Minimum Assertion Width (SLP_S3_MIN_ASST_WDTH):** This 2-bit value indicates the minimum assertion width of the PMU_SLP_S3_N signal to guarantee that the Main power supplies have been fully power-cycled.<br>This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc.<br>Settings are:<br>• 00: 60 µsec<br>• 01: 1 ms<br>• 10: 50 ms<br>• 11: 2 sec<br>This field is not writable when the SLP_Sx# Stretching Policy Lock- Down bit is set.<br>This bit is cleared by the pri_pwrgood_rst_b pin. |
| 9 | 0x0<br>RW/1C/V | **Host Reset Status (HOST_RST_STS):** This bit is set by hardware when a host partition reset (not a global reset, DeepSx, or G3) occurs. |
| 8 | 0x0<br>RO | **Reserved (RSVD_8):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RW | **SWSMI Rate Select (SWSMI_RATESEL):** This 2-bit value indicates when the SWSMI timer will time out.<br>Valid values are:<br>• 00 1.5ms ± 0.6ms<br>• 01 16ms ± 4ms<br>• 10 32ms ± 4ms<br>• 11 64ms ± 4ms<br>This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 5:4 | 0x0<br>RW/L | **SLP_S4# Minimum Assertion Width (S4MAW):** This 2-bit value indicates the minimum assertion width of the SLP_S4# signal to guarantee that the DRAMs have been safely power-cycled.<br>This value may be modified per platform depending on DRAM types, power supply capacitance, etc.<br>Valid values are:<br>• 11: 1 second<br>• 10: 2 seconds<br>• 01: 3 seconds<br>• 00: 4 seconds<br>This value is used in two ways:<br>• 1. If the PMU_SLP_S45_N assertion width is ever shorter than this time, the status bit MS4V of the GEN_PMCON_A register is set for BIOS to read when S0 is entered<br>• 2. If enabled by bit 3 in this register, the hardware will prevent the PMU_SLP_S45_N signal from deasserting within this minimum time period after asserting. |
| 3 | 0x0<br>RW/L | **SLP_S4# Assertion Stretch Enable (S4ASE):**<br>When set to 1, the PMU_SLP_S45_N pin (which includes the ME override logic) will minimally assert for the time specified in bits 5:4 of this register.<br>When 0, the minimum assertion time for PMU_SLP_S45_N is the same as the timing defined in the timing tables. |
| 2 | 0x1<br>RW | **RTC_PWR_STS (RPS):** The SoC will set this bit to 1 to indicate a weak or missing battery. The bit will remain set until the software clears it by writing a 0 back to this bit position. This bit is in the RTC power well and is not cleared by any type of reset. |
| 1 | 0x1<br>RW/1C | **PWR_FLR (PF):**<br>• 0 = Indicates that the trickle current has not failed since the last time the bit was cleared.<br>• 1 = Indicates that the trickle current (from the main battery or trickle supply) was removed or failed.<br>Software writes a 1 to this bit to clear it. |
| 0 | 0x1<br>RW/V | **AFTERG3_EN (AG3E):** Determines what state to go to when power is reapplied after a power failure (G3 state).<br>• 0 = System will return to an S0 state (boot) after power is re-applied.<br>• 1 = System will return to the S5 state. In the S5 state, the only enabled wake-up event is the Power Button or any enabled wake event that was preserved through the power failure. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |

## 58.2.6 Extended Test Mode Register 3 (ETR3)—Offset ACh

This register resides in the primary Core well. All bits except bit[23:16] are reset by host_deep_rst_b. Bit[23:16] are reset by pri_pwrgood_rst_b only.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:2] + ACh

**Default:** 80000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x1<br>RW/V/L | **CF9h Lockdown (CF9LOCK):** When set, this will lock the "CF9h Global Reset" bit.<br>When set, this register locks itself. This register is reset by a CF9h reset.<br><br>***Note:*** In manufacturing/ debug environments this bit can be left as default 0b. In all other environments, BIOS must program this bit to 1b. |
| 30:21 | 0x0<br>RO | **Reserved (RSVD_30_21):** These bits are reserved for Intel. |
| 20 | 0x0<br>RW/L | **CF9h Global Reset (CF9GR):**<br>When this bit is set, a CF9h write of 6h or Eh will cause a Global Reset of both the Host and the ME partitions.<br>If this bit is cleared, a CF9h write of 6h or Eh will only reset the Host partition. |
| 19:0 | 0x0<br>RO | **Reserved (RSVD_19_0):** These bits are reserved for Intel. |

## 58.3 Registers in Memory Space—PWRMBASE

### 58.3.1 Wake Alarm Device Timer: AC (WADT_AC)—Offset 0h

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well and SUS well are down, they are marked as DSW bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 0h

**Default:** FFFFFFFFh

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xFFFFFFFF RW/V | **Wake Alarm Device Timer Value for AC Mode (WADT_AC_VAL):** This field contains the 32-bit wake alarm device timer value (granularity 1s) for AC power.<br>The timer begins decrementing when written to a value other than FFFFFFFFh (regardless of the power source when the write occurs).<br>Upon counting down to 0:<br>• If on AC power, GPE0b_STS.WADT_STS will be set. This status bit being set will generate a host wake if GPE0b_EN.WADT_EN is 1.<br>• If the power source is DC at this time, the status bit is not set. However, if AC power subsequently returns to the platform, the AC Expired Timer begins running. See the WADT_EXP_AC register for details.<br>• The timer returns to its default value of FFFFFFFFh. |

## 58.3.2 Wake Alarm Device Timer: DC (WADT_DC)—Offset 4h

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well and SUS well are down, they are marked as DSW bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 4h

**Default:** FFFFFFFFh

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xFFFFFFFF RW/V | **Wake Alarm Device Timer Value for DC Mode (WADT_DC_VAL):** This field contains the 32-bit wake alarm device timer value (granularity 1s) for DC power. The timer begins decrementing when written to a value other than FFFFFFFFh (regardless of the power source when the write occurs). Upon counting down to 0:<br>• If on DC power, GPE0b_STS.WADT_STS will be set. This status bit being set will generate a host wake if GPE0b_EN.WADT_EN is 1.<br>• If the power source is AC at this time, the status bit is not set. However, if DC power subsequently returns to the platform, the DC Expired Timer begins running. See the WADT_EXP_DC register for details.<br>• The timer returns to its default value of FFFFFFFFh. |

### 58.3.3 Wake Alarm Device Expired Timer: AC (WADT_EXP_AC)—Offset 8h

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well and SUS well are down, they are marked as DSW bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 8h

**Default:** FFFFFFFFh

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xFFFFFFFF RW/V | **Wake Alarm Device Expired Timer Value for AC Mode (WADT_EXP_AC_VAL):** This field contains the 32-bit wake alarm device "Expired Timer" value (granularity 1s) for AC power.<br>The timer begins decrementing after switching from DC to AC power, in the case where the WADT_AC timer has already expired while the platform was on DC power.<br>This timer only decrements while operating on AC power. So if the power source switches back to DC power, the timer will stop (but not reset).<br>When AC power returns, the timer will again begin decrementing.<br>***Note:*** This timer will only begin decrementing under the conditions described above if this field has been configured for something other than its default value of FFFFFFFFh (i.e. FFFFFFFFh = disabled).<br>Upon expiration of this timer:<br>• If on AC power, GPE0b_STS.WADT_STS will be set. This status bit being set will generate a host wake if GPE0b_EN.WADT_EN is 1.<br>• BOTH the AC and DC Expired Timers return to their defaults value of FFFFFFFFh. |

### 58.3.4 Wake Alarm Device Expired Timer: DC (WADT_EXP_DC)—Offset Ch

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well and SUS well are down, they are marked as DSW bits.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + Ch

**Default:** FFFFFFFFh

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xFFFFFFFF RW/V | **Wake Alarm Device Expired Timer Value for DC Mode (WADT_EXP_DC_VAL):** This field contains the 32-bit wake alarm device "Expired Timer" value (granularity 1s) for DC power.<br>The timer begins decrementing after switching from AC to DC power, in the case where the WADT_DC timer has already expired while the platform was on AC power.<br>This timer only decrements while operating on DC power. So if the power source switches back to AC power, the timer will stop (but not reset).<br>When DC power returns, the timer will again begin decrementing.<br>***Note:*** This timer will only begin decrementing under the conditions described above if this field has been configured for something other than its default value of FFFFFFFFh (i.e. FFFFFFFFh = disabled).<br>Upon expiration of this timer:<br>• If on DC power, GPE0b_STS.WADT_STS will be set. This status bit being set will generate a host wake if GPE0b_EN.WADT_EN is 1.<br>• BOTH the AC and DC Expired Timers return to their defaults value of FFFFFFFFh. |

## 58.3.5 Power and Reset Status (PRSTS)—Offset 10h

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well is down, they are marked as suspend well bits. All suspend well bits in this register are reset by global_rst_b.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 10h

**Default:** 05000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x5<br>RO/V | **Power Management Controller Product ID (PMC_PRODID):** This field communicates the Product Family of the power management functionality. |
| 23:16 | 0x0<br>RO/V | **Power Management Controller Revision ID (PMC_REVID):** This field communicates the implementation revision of the power management functionality. |
| 15:12 | 0x0<br>RO | **Reserved (RSVD_15_12):** Reserved. |
| 11 | 0x0<br>RW/1C/V | **Patch Copied Over Status (PATCH_COPIED_STS):** The SoC sets this bit when the Patch code is successfully authenticated and loaded into the SoC from the flash. |
| 10 | 0x0<br>RO/V | **Patch Revision Not Compatible Status (PATCH_REV_NOT_COMP):** The SoC sets this bit when the Patch code is not compatible with the Power Management Controller Product and/or Revision.<br>This bit remains valid after a PMC Patch load is attempted until the next global_rst_b. |
| 9 | 0x0<br>RO/V | **Patch Load Timeout Status (PATCH_LOAD_TO):** The SoC sets this bit if the loading function fails to complete within a reasonable time limit. |
| 8 | 0x0<br>RO/V | **PMC Operational Status (PMC_OP_STS):** The SoC sets this bit when the PMC becomes operational after completing the Patch Load, or, if no patch is needed.<br>BIOS must wait for this bit to be set before performing resets or sleep events.<br>This bit remains valid after a PMC Patch load until the next global_rst_b. |
| 7 | 0x0<br>RW/1C/V | **VE Watch Dog Timer Status (VE_WDT_STS):** This bit will be set to '1' when the VE Watch Dog Timer triggers a reset.<br>It will be cleared by a write of '1' by software. |
| 6 | 0x0<br>RW/1C/V | **IE Host Wake Status (IE_HOST_WAKE_STS):** This bit is set when the IE generates a non-maskable wake event and is not affected by any other enable bit.<br>When this bit is set, the host power management logic wakes to S0. |
| 5 | 0x0<br>RW/1C/V | **Wake On LAN Override Wake Status (WOL_OVR_WK_STS):** This bit gets set when integrated LAN Signals a Power Management Event AND the system is in S5. BIOS can read this status bit to determine this wake source. Software clears this bit by writing a 1 to it. |
| 4 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 3 | 0x0<br>RO/V | **HR_SHAD_IE_EN_STS:** This register shadows the IE ROM Check in host space.<br>• 0 = IE is disabled<br>• 1 = IE is enabled |
| 2:1 | 0x0<br>RO | **Reserved (RSVD_2_1):** Reserved. |
| 0 | 0x0<br>RW/1C/V | **ME_HOST_WAKE_STS:** This bit is set when the ME generates a non-maskable wake event and is not affected by any other enable bit.<br>When this bit is set, the host power management logic wakes to S0. |

## 58.3.6 Power Management Configuration Reg 1 (PM_CFG)—Offset 18h

This register contains misc. fields used to configure the SoC power management behavior. This register is in multiple power wells and reset domains (see below).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 18h

**Default:** 00000020h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **SATA Interface Disable Indication (SATA_DIS_IND):**<br>When set to 1, this bit indicates that the SATA interface on the SoC is completely disabled. Therefore, the mPhy lanes normally assigned to SATA will operate through the PCIe controller (or not be used at all).<br>When cleared, it indicates that at least some of the SATA interface lanes may be used actively in the SoC by the SATA controller, so none of the four lanes assigned to SATA interface can operate through the PCIe controller. |
| 30 | 0x0<br>RW | **Timing t591 (TIMING_T591):** This field configures t591 timing involved in the over-clocking flow (Wait times between writes from PMC to ICC registers when executing BCLK slow ramp flow)<br>***Note:*** Encodings are all min timings. |
| 29 | 0x0<br>RW | **Allow 24MHz Crystal Oscillator Shutdown (ALLOW_24_OSC_SD):**<br>When this bit is '0', the 24MHz crystal oscillator will always be running while in S0.<br>When this bit is '1', the 24MHz crystal oscillator may be shut down in S0 (Cx only) if all other conditions allow. |
| 28 | 0x0<br>RW | **Time Sync Maximum Attempts (TS_MAXTRY):** The value of this bit determines how many times the PMC will retry time synchronization before stopping and reporting failure to the initiator.<br>Encoding:<br>• 0 Maximum of 5 attempts<br>• 1 Maximum of 10 attempts |
| 27 | 0x0<br>RW/L | **Debug Mode Lock (DBG_MODE_LOCK):** When this bit is set to '1', the following bit is locked down<br>• ETR3.SKIP_HOST_RST_HS<br>• ETR3.RST_SX_TO_DBG<br>• ETR3.PSMI_DGB_MODE_EN<br>This bit becomes locked when a value of 1b is written to it. Writes to 0 to this bit are always ignored. Once locked, the only way the bit gets cleared is through a PLTRST# assertion. |
| 26:24 | 0x0<br>RO | **Reserved (RSVD_26_24):** Reserved. |
| 23 | 0x0<br>RW | **Enable Global Reset on Uncorrectable Parity Error on PMC SRAM Interface (EN_PMC_UNC_ERR):** If this bit is set to '1', a global reset is triggered when a parity error is detected on PMC's SRAM interfaces.<br>No action is taken on such parity errors if this bit is '0'. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 22 | 0x0<br>RO | **Reserved (RSVD_22):** Reserved. |
| 21 | 0x0<br>RW | **RTC Wake from DeepSx Disable (RTC_DSX_WAKE_DIS):** When set, this bit disables RTC wakes from waking the system from DeepSx. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 20 | 0x0<br>RO | **Reserved (RSVD_20):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19:18 | 0x0<br>RW/L | **SLP_SUS# Minimum Assertion Width (SLP_SUS_MIN_ASST_WDTH):** This 2-bit value indicates the minimum assertion width of the SLP_SUS# signal to guarantee that the SUS well power supplies have been fully power-cycled.<br>This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc.<br>Settings are:<br>• 00 = 0 ms (i.e. stretching disabled - default)<br>• 01 = 500ms<br>• 10 = 1s<br>• 11 = 4s<br>This field is not writable when the SLP_Sx# Stretching Policy Lock- Down bit is set.<br>This field is ignored when exiting a G3 state if the "Disable SLP_X Stretching After SUS Power Failure" bit is set. Note that unlike with all other SLP_* pin stretching, this disable bit only impacts SLP_SUS# stretching during G3 exit rather than both G3 and DeepSx exit. SLP_SUS# stretching always applies to DeepSx regardless of the disable bit.<br>**Programming Note:** For platforms that enable DeepSx, BIOS must program SLP_SUS# stretching to be greater than or equal to the largest stretching value on any other SLP_* pin (SLP_S3#, SLP_S4#, SLP_LAN#, or SLP_A#).<br>This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 17:16 | 0x0<br>RW/L | **SLP_A# Minimum Assertion Width (SLP_A_MIN_ASST_WDTH):** This 2-bit value indicates the minimum assertion width of the SLP_A# signal to guarantee that the ASW power supplies have been fully power-cycled.<br>This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc.<br>Settings are:<br>00 = 0 ms (i.e. stretching disabled - default)<br>01 = 4 s<br>10 = 98 ms<br>11 = 2 s<br>This field is not writable when the SLP_Sx# Stretching Policy Lock- Down bit is set.<br>This field is ignored when exiting a G3 or DeepSx state if the "Disable SLP_X Stretching After SUS Power Failure" bit is set.<br>This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 15:14 | 0x0<br>RW/L | **SLP_LAN# Minimum Assertion Width (SLP_LAN_MIN_ASST_WDTH):** This 2-bit value indicates the minimum assertion width of the SLP_LAN# signal to guarantee that the power to the PHY has been fully power-cycled.<br>This value may be modified per platform depending on power supply, capacitance, board capacitance, power failure detection circuits, etc.<br>This field is not writable when the SLP_Sx# Stretching Policy Lock Down bit is set to '1'. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 13 | 0x0<br>RW | **After G3 Last State Enable (AG3_LS_EN):** When GEN_PMCON_B.AG3E is 0, AG3_LS_EN determines whether the SoC will consider the platforms previous state when determining whether to power-up after G3. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active.<br>Encodings:<br>• 0 (default): SoC power-up policies after G3 do not depend on the platforms state when the G3 occurred.<br>• 1: SoC power-up policies after G3 depend on the platforms state when the G3 occurred.<br>- If the power failure occurred while the platform was in S0 or while in the process of going to S0, the platform will power-up to S0 upon exiting G3.<br>- If the power failure occurred while the platform was in Sx or while in the process of going to Sx, the platform will stay in S4/S5 upon exiting G3.<br>***Note:*** This bit applies only when GEN_PMCON_B.AG3E is 0. If AG3E is 1, the platform will always stay in S4/S5 after G3 regardless of the value of AG3_LS_EN.<br>***Note:*** The destination state is considered to be S0 during both host wake and throughout host partition resets with/without power cycle. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 12 | 0x0<br>RW | **After Type 8 Global Reset Last State Enable (A8GR_LS_EN):** AGR_LS_EN determines whether the SoC will consider the platforms previous state when determining whether to power-up after non-thermal and non-explicitly requested type 8 global resets. See the Reset Sources table for a list of impacted reset types. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active.<br>Encodings:<br>• 0 (default): SoC power-up policies after a global reset do not depend on the platforms state when the reset occurred.<br>• 1: SoC power-up policies after a global reset depend on the platforms state when the reset occurred.<br>- If the global reset occurred while the platform was in S0 or while in the process of going to S0, the platform will power-up to S0 after the reset.<br>- If the global reset occurred while the platform was in Sx or while in the process of going to Sx, the platform will stay in S5 upon exiting G3.<br>***Note:*** The destination state is considered to be S0 during both host wake and throughout host partition resets with/without power cycle. |
| 11 | 0x0<br>RW | **Global Reset Three Strike Counter Enable (GR_TSC_EN):** When set, GR_TSC_EN will cause the PMC to keep the platform in S5 after the third consecutive type 7 global reset occurs during the boot flow. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active.<br>The three strike counter is reset in the following situations:<br>• The system reaches S0<br>• A type 8 global reset occurs, including after the three strike counter causes the system to stay in S5.<br>• RSMRST_N asserts (due to DeepSx entry or a G3 event occurring) |
| 10 | 0x0<br>RW | **Power Button Debounce Mode (PB_DB_MODE):** This bit controls when interrupts (SMI#, SCI) are generated in response to assertion of the PMU_PWRBTN_N pin. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active.<br>This bit's values cause the following behavior:<br>• '0': The 16ms debounce period applies to all usages of the PMU_PWRBTN_N pin (legacy behavior).<br>• '1': When a falling edge occurs on the PMU_PWRBTN_N pin, an interrupt is generated and the 16ms debounce timer starts. Subsequent interrupts are masked while the debounce timer is running.<br>***Note:*** Power button override logic always samples the post-debounce version of the pin. |
| 9:8 | 0x0<br>RW/L | **Reset Power Cycle Duration (PWR_CYC_DUR):** The value in this register determines the minimum time a platform will stay in reset (SLP_S3#, SLP_S4#, SLP_S5# asserted and also SLP_A# and SLP_LAN# asserted if applicable) during a host partition reset with power cycle or a global reset.<br>The duration programmed in this register takes precedence over the applicable SLP_# stretch timers in these reset scenarios. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active.<br>This field is not writable when the SLP_Sx# Stretching Policy Lock-Down bit is set.<br>Note that the duration programmed in this register should never be smaller than the stretch duration programmed in the following registers<br>• GEN_PMCON_3.SLP_S3_MIN_ASST_WDTH<br>• GEN_PMCON_3.S4MAW<br>• PM_CFG.SLP_A_MIN_ASST_WDTH<br>• PM_CFG.SLP_LAN_MIN_ASST_WDTH |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_7_6):** Reserved. |
| 5 | 0x1<br>RW/V | **CPU OC Strap (COCS):** SW programs this pin with the value that should be reflected to the GPIO8_OCS pin, when the pin is in native mode.<br>Hardware also sets this bit when the over-clocking watchdog timer expires. |
| 4:2 | 0x0<br>RO | **Reserved (RSVD_4_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1:0 | 0x0<br>RW/V | **Timing t581 (TIMING_T581):** This field configures the t581 timing involved in the power down flow (CPUPWRGD inactive to ICC_ICLK_INIT inactive). Encodings (all min timings):<br>• 00: 10 us (default)<br>• 01: 100 us<br>• 10: 1 ms<br>• 11: 10 ms<br>reset_type=host_deep_rst_b |

### 58.3.7 PCH Power Management Status (PCH_PM_STS)—Offset 1Ch

This register contains misc. fields used to record events pertaining to SoC power management. Unless otherwise indicated, all RWC bits are cleared with a write of 1 by software. This register is in multiple power wells and reset domains (see below).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 1Ch

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved (RSVD_30_25):** Reserved. |
| 16 | 0x0<br>RW/1C/V | **ADR Reset Status (ADR_RST_STS):** When set to '1', this bit indicates that the Asynchronous DRAM Refresh (ADR) flow was executed during a prior global reset entry. |
| 15:13 | 0x0<br>RW | **BIOS Discovered DDR Speed (SPD_DDR_SPEED):**<br>• 000=DDR 1600<br>• 001=DDR 1867<br>• 010=DDR 2133<br>• 011=DDR 2400<br>• 100=DDR 2666 |
| 12 | 0x0<br>RW | **Equal Speed Enable (EQUAL_SPEED_ENABLE):**<br>• 0= Do not enable SSC PLL relock when SPD_DDR_SPEED = MAX_DDR_SKU<br>• 1= enable SSC PLL relock when SPD_DDR_SPEED = MAX_DDR_SK |
| 11 | 0x0<br>RW | **The triggered reset is an SPD reset (SPD_BIOS_RESET):**<br>• 0= not SPD reset<br>• 1= SPD Reset |
| 10:7 | 0x0<br>RO | **Reserved (RSVD_10_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **IE uncorrectable Error Status (IE_UE_STS):** PMC FW Sets this bit as an indication that ME uncorrectable error as been forwarded to PMC FW |
| 5:1 | 0x0<br>RO | **Reserved (RSVD_5_1):** Reserved. |
| 0 | 0x0<br>RW/1C/V | **ME Uncorrectable Error Status (ME_UE_STS):** PMC FW sets this bit as an indication that a ME uncorrectable error has been forwarded to PMC FW. |

## 58.3.8 PCH Power Management Status (PCH_PM_STS2)—Offset 24h

This register contains misc. fields used to record events pertaining to SoC power management. Unless otherwise indicated, all RWC bits are cleared with a write of '1' by software.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 24h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_31_21):** Reserved. |
| 20:17 | 0x0<br>RO/V | **Current SPD SPEED (CURRENT_SPD_SPEED):**<br>0xxxx = undefined, use max DDR mem config (fuse)<br>• 1000 = DDR1600<br>• 1001 = DDR1866<br>• 1010 = DDR2133<br>• 0000 = DDR2400<br>***Note:*** The value of current SPD SPEED is only updated when the SPD reset is happened. |
| 16 | 0x0<br>RO/V | **SPD_RESET_PCODE:**<br>• 0 = PMC indicates to Pcode that this is not a SPD Reset.<br>• 1 = PMC indicates to Pcode that this is a SPD Reset<br>Set by PMC when SPD Reset is triggered |
| 15 | 0x0<br>RW/1C/V | **IE Host Boot Prep Done Failure (IE_HBPD):** IE Host Boot Preparation did not complete. |
| 14 | 0x0<br>RW/1C/V | **CPU Reset Done Failure (CRD):** CPU Reset Done message did not arrive from the CPU. |
| 13 | 0x0<br>RW/1C/V | **ME Host Boot Prep Done Failure (ME_HBPD):** Intel® ME Host Boot Preparation did not complete. |
| 12 | 0x0<br>RW/1C/V | **PINSTOP Acknowledge Failure (PINSTOP_ACK):** The GbE PHY did not respond to the PINSTOP message. |
| 11 | 0x0<br>RW/1C/V | **PSCHS Acknowledge Failure (PSCHS_ACK):** The GbE MAC did not respond to the PSCHS message. |
| 10 | 0x0<br>RW/1C/V | **SMT Reset Acknowledge Failure (SMT_RST_ACK):** One or more SMT controllers did not respond to the Intel® Management Engine bus reset warning. |
| 9 | 0x0<br>RW/1C/V | **EXI State Transition Acknowledge Failure (EXI_STATE_TRANS_ACK):** EXI state transition ACK did not arrive. |
| 8 | 0x0<br>RW/1C/V | **SPI Common Prep Handshake Failure (SPI_HRHS):** The SPI controller did not complete the host partition reset/Sx entry handshake. |
| 7 | 0x0<br>RW/1C/V | **XCK Common Prep Handshake Failure (XCK_HRHS):** The integrated clocking unit did not complete the host partition reset/Sx entry handshake. |
| 6 | 0x0<br>RW/1C/V | **CPU S345/Reset Warn Acknowledge Failure (CPU_S345RW_ACK):** The CPU did not respond to the GO_S345 or RESET_WARN message. |
| 5 | 0x0<br>RW/1C/V | **CPU S1 Acknowledge Failure (CPU_S1_ACK):** The CPU did not respond to the GO_S1_XXX message. |
| 4 | 0x0<br>RW/1C/V | **DMI L23 Entry Failure (DMI_L23):** The DMI interface did not respond to the request to entry L23. |
| 3 | 0x0<br>RW/1C/V | **SMBus Host Reset Handshaking Failure (SMB_SRHS):** The host SMBus controller did not complete the host partition reset handshake. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C/V | **South Port L23 Entry Failure (SP_L23):** The SoC PCI Express ports did not complete the host partition reset/Sx entry handshake. |
| 1 | 0x0<br>RW/1C/V | **XHCI Common Prep Handshake Failure (XHCI_HRHS):** The USB XHCI controller did not complete the host partition reset/Sx entry handshake. |
| 0 | 0x0<br>RW/1C/V | **EVA L23 Entry Failure (EVA_L23):** The EVA did not complete the host partition reset/Sx entry handshake. |

## 58.3.9 ADR Enable (ADR_EN)—Offset F0h

This register contains several enable bits related to the ”Automatic DIMM Self-Refresh” Asynchronous DRAM Refresh (ADR) feature. The default value of each bit is chosen based on the expected usage, but the values are programmable to retain flexibility. See the section on ADR later in chap. 16 for more information. This register is in the CORE power well and is reset by host_prim_rst_b.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + F0h

**Default:** 200F0DE4h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_31_30):** Reserved. |
| 29 | 0x1<br>RW | **ADR GPIO Reset Enable (ADR_GPIO_RST_EN):** If this bit is set to '1', a Global Reset will be generated once the internal flush sequence is completed after:<br>• Triggered by the ADR_TRIGGER signal pin (if available in the SoC product SKU).<br>-OR-<br>• Qualified internal events and the following are true:<br>The ADR feature is part of the particular SoC product SKU, and<br>ADR_FEAT_EN (bit 0) is a 1.<br>If this bit is cleared, the active-high ADR_TRIGGER signal pin can still be enabled to affect the internal flush, but on its own, the signal does not trigger a Global Reset.<br>***Note:*** Unlike the other ADR_EN bits in this register, this enable bit is not required to be set to 1 for this source to trigger the warning to the SoC and CPU. This bit only impacts whether or not this source will actually trigger a Global Reset. |
| 28:20 | 0x0<br>RO | **Reserved (RSVD_28_20):** Reserved. |
| 19 | 0x1<br>RW/V | **IE Uncorrectable Error ADR Enable (IE_UERR_ADR_EN):** If this bit is set to '1', the IE uncorrectable SRAM error is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 18 | 0x1<br>RW/V | **IE Initiated Global Reset ADR Enable (IEGBL_ADR_EN):** If this bit is set to '1', the IE-initiated global reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 17 | 0x1<br>RW/V | **IE FW Watchdog Timer ADR Enable (IEWDT_ADR_EN):** If this bit is set to '1', the IE firmware watchdog timer global reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 16 | 0x1<br>RW/V | **IE Initiated Power Button Override ADR Enable (IEPBO_ADR_EN):** If this bit is set to '1', the IE-Initiated Power Button Override Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 15 | 0x0<br>RO | **Reserved (RSVD_15):** Reserved. |
| 14 | 0x0<br>RW/V | **PMC Parity Error ADR Enable (PMC_PARERR_ADR_EN):** If this bit is set to '1', the PMC SUS RAM Parity Error Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '0' because this reset source is considered to be time-critical. |
| 13 | 0x0<br>RO | **Reserved (RSVD_13):** Reserved. |
| 12 | 0x0<br>RW/V | **SYS_PWROK Failure ADR Enable (SYSPWR_ADR_EN):** If this bit is set to '1', the SYS_PWROK Failure Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '0' because this reset source is considered to be time-critical. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 11 | 0x1<br>RW/V | **ME Uncorrectable Error ADR Enable (ME_UERR_ADR_EN):** If this bit is set to '1', the Intel® ME Uncorrectable SRAM error Global Reset source is included in the ADR feature.<br>***Note:*** Note: The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 10 | 0x1<br>RW/V | **PMC FW Watchdog Timer ADR Enable (PMCWDT_ADR_EN):** If this bit is set to '1', the PMC Firmware Watchdog Timer Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 9 | 0x0<br>RO | **Reserved (RSVD_9):** Reserved. |
| 8 | 0x1<br>RW/V | **ME-Initiated Global Reset ADR Enable (MEGBL_ADR_EN):** If this bit is set to '1', the ME-initiated Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 7 | 0x1<br>RW/V | **ME FW Watchdog Timer ADR Enable (MEWDT_ADR_EN):** If this bit is set to '1', the ME Firmware Watchdog Timer Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 6 | 0x1<br>RW/V | **ME-Initiated Power Button Override ADR Enable (MEPBO_ADR_EN):** If this bit is set to '1', the ME-Initiated Power Button Override Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 5 | 0x1<br>RW/V | **Power Button Override ADR Enable (PBO_ADR_EN):** If this bit is set to '1', the Power Button Override Global Reset source is included in the ADR feature.<br>***Note:*** The default value for this bit was chosen to be '1' because this reset source is not considered to be time-critical. |
| 4:3 | 0x0<br>RO | **Reserved (RSVD_4_3):** Reserved. |
| 2 | 0x1<br>RW/V | **PMC FW-Initiated Global Reset ADR Enable (PMCGBL_ADR_EN):** If this bit is set to '1', all PMC firmware-initiated Global Reset sources are included in the ADR feature. The specific sources are:<br>- SMBUS slave unconditional power down message<br>• I/O space CF9h write or Shutdown Special-Cycle with the policy set for Global Reset.<br>• Host partition reset entry timeout.<br>• Sx entry timeout.<br>• Any Host partition reset triggered with the "Promote Host Partition Reset to Global Reset" policy bit set to 1.<br>***Note:*** The default value for this bit was chosen to be '1' because any FW-handled global reset source is not considered to be time-critical. |
| 1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 0 | 0x0<br>RW/V | **ADR Feature Enable (ADR_FEAT_EN):** If this bit is a '1' and the SoC product SKU supports ADR, SoC, support for Asynchronous DRAM Refresh is enabled. The other bits in this register determine which individual reset sources are included. |

## 58.3.10 System Time Capture Trigger (SYSTIMCT)—Offset F8h

This register contains bits related to the "Audio Time Capture (ATC)" feature. This register is in the CORE power well and is reset by host_prim_rst_b.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + F8h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD_31_1):** Reserved. |
| 0 | 0x0<br>RW/1S/V | **Capture Now (CAPTURE_NOW):** Write of '1' to this bit triggers time capture in all components supporting this feature.<br>Hardware clears this register to indicate that time capture is complete.<br>Once the bit is set to '1', subsequent writes of '0' or '1' have no effect until the bit is cleared by hardware.<br>Please refer to the Audio Time Synchronization section for more details on Audio Synchronization and for a list of consumers of the "capture" signal from PMC.<br>Implementation note: Once the bit is written to '1'. hardware should keep the "capture" signal asserted high for 16 PMC clocks, then keep the signal deasserted low for another 16 PMC clocks and then clear this register bit. |

## 58.3.11 ACPI Timer Control (ACPI_TMR_CTL)—Offset FCh

This register allows software to disable the ACPI Timer, which could result in power savings for the SoC. This register is in the CORE power well and is reset by host_prim_rst_b.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + FCh

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved (RSVD_31_2):** Reserved. |
| 1 | 0x0<br>RW | **ACPI Timer Disable (ACPI_TIM_DIS):** This bit determines whether the ACPI Timer is enabled to run. Note that even when enabled, the timer only runs during S0.<br>- 0: ACPI Timer is enabled (default)<br>- 1: ACPI Timer is disabled (halted at the current value) |
| 0 | 0x0<br>RW/1S/V | **ACPI Timer Clear (ACPI_TIM_CLR):** Writing a 1 to this bit will clear the ACPI Timer to all 0s. Hardware will automatically clear the bit back to '0' once the timer clear operation has completed. Writing a 0 to this bit has no effect.<br>Implementation Note: The SoC must be capable of honoring this bit even while ACPI_TIM_DIS=1. |

## 58.3.12 GPIO Configuration (GPIO_CFG)—Offset 120h

This register is in the PRIMARY power well and is reset by global_rst_b. reset_type=global_rst_b.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 120h

**Default:** 00000432h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:11 | 0x0<br>RO | **Reserved (RSVD_31_11):** Reserved. |
| 10:8 | 0x4<br>RW | **GPIO Group to GPE_DW2 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64].<br>• 0h = reserved[7:0] and GPP_A[23:0]<br>• 1h = reserved[7:0] and GPP_B[23:0]<br>• 2h = reserved[7:0] and GPP_C[23:0]<br>• 3h = reserved[7:0] and GPP_D[23:0]<br>• 4h = reserved[7:0] and GPP_E[23:0]<br>• 5h = reserved[7:0] and GPP_F[23:0]<br>• 6h = reserved[23:0] and GPP_G[7:0]<br>• 7h = reserved[20:0] and GPD[11:0] |
| 7 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 6:4 | 0x3<br>RW | **GPIO Group to GPE_DW1 assignment encoding (GPE0_DW1):** This register assigns a specific GPIO Group to the ACPI GPE0[63:32].<br>• 0h = reserved[7:0] and GPP_A[23:0]<br>• 1h = reserved[7:0] and GPP_B[23:0]<br>• 2h = reserved[7:0] and GPP_C[23:0]<br>• 3h = reserved[7:0] and GPP_D[23:0]<br>• 4h = reserved[7:0] and GPP_E[23:0]<br>• 5h = reserved[7:0] and GPP_F[23:0]<br>• 6h = reserved[23:0] and GPP_G[7:0]<br>• 7h = reserved[20:0] and GPD[11:0] |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 2:0 | 0x2<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** This register assigns a specific GPIO Group to the ACPI GPE0[31:0].<br>• 0h = reserved[7:0] and GPP_A[23:0]<br>• 1h = reserved[7:0] and GPP_B[23:0]<br>• 2h = reserved[7:0] and GPP_C[23:0]<br>• 3h = reserved[7:0] and GPP_D[23:0]<br>• 4h = reserved[7:0] and GPP_E[23:0]<br>• 5h = reserved[7:0] and GPP_F[23:0]<br>• 6h = reserved[23:0] and GPP_G[7:0]<br>• 7h = reserved[20:0] and GPD[11:0] |

## 58.3.13 Global Reset Causes (GBLRST_CAUSE0)—Offset 124h

This register logs causes of host partition resets.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 124h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:25 | 0x0<br>RO | **Reserved (RSVD_31_25):** Shadow additional HW initiated GBLRST causes once they're added. |
| 24 | 0x0<br>RW/1C/V | **IE HW Uncorrectable Error (IE_UNCOR_ERR):** This bit is set to '1' by hardware when a global reset is triggered by IE hardware due to the detection of an uncorrectable ECC or parity error on a data read from one of its SRAMs. |
| 23 | 0x0<br>RW/1C/V | **IE FW Watchdog Timer (IE_WDT):** This bit is set to '1' by hardware when a global reset is triggered by the second expiration of the IE firmware watchdog timer. |
| 22 | 0x0<br>RW/1C/V | **IE-Initiated Global Reset (IE_GBL):** This bit is set to '1' by hardware when a global reset is triggered by an ME FW write of 1's to both GENCTL.'IE-Partition Reset' and GENCTL.'IE-Initiated Host Reset With Power Cycle' in the same write cycle (this is IE FW's method of requesting a global reset). |
| 21 | 0x0<br>RW/1C/V | **IE-Initiated Power Button Override (IE_PBO):** This bit is set to '1' by hardware when a global reset is triggered by an IE FW write of '1' to GENCTL.'IE-Initiated Power Button Override |
| 20 | 0x0<br>RW/1C/V | **Over-Clocking WDT Expiration In ICC Survivability Mode (OC_WDT_EXP_ICCSURV):** This bit is set to '1 by hardware when a global reset is triggered by the expiration of the over-clocking watchdog timer while running in a mode that has ICC survivability impact (OC_WDT_ICCSURV=1). |
| 19 | 0x0<br>RW/1C/V | **Over-Clocking WDT Expiration In Non-ICC Survivability Mode (OC_WDT_EXP_NO_ICCSURV):** This bit is set to '1 by hardware when a global reset is triggered by the expiration of the over-clocking watchdog timer while running in a mode that does not have ICC survivability impact (OC_WDT_ICCSURV=0). |
| 18 | 0x0<br>RW/1C/V | **ADR GPIO Reset (ADR_GPIO_RST):** This bit is set to '1' by hardware when a global reset is triggered by the assertion of the SoC ADR_TRIGGER signal pin. |
| 17 | 0x0<br>RW/1C/V | **ME HW Uncorrectable Error (ME_UNCOR_ERR):** This bit is set to '1' by hardware when a global reset is triggered by ME hardware due to the detection of an uncorrectable ECC or parity error on a data read from one of its SRAM s. |
| 16 | 0x0<br>RW/1C/V | **CPU Thermal Runaway Watchdog Timer (CPU_THRM_WDT):** This bit is set to '1' by hardware when a global reset is triggered by the expiration of the CPU Thermal Runaway Watchdog Timer. |
| 15:14 | 0x0<br>RW/1C/V | **Reserved (RSVD_15_14):** Reserved. |
| 13 | 0x0<br>RW/1C/V | **AS Well Power Failure (ASW_FLR):** This bit is set to '1' by hardware when a global reset is triggered by an unexpected loss of ASW power (i.e. a deassertion of APWROK at an unexpected time). |
| 12 | 0x0<br>RW/1C/V | **SYS_PWROK Failure (SYSPWR_FLR):** This bit is set to '1' by hardware when a global reset is triggered by an unexpected loss of SYS_PWROK. FW arms this global reset source via GBLRST_CTL.EN_SYSPWR_FLR. This register bit is in the RTC power well and is retained if the system has a functioning RTC battery during the G3 Mechanical Off state. |
| 11 | 0x0<br>RW/1C/V | **PCH_PWROK Failure (PCHPWR_FLR):** This bit is set to '1' by hardware when a global reset is triggered by an unexpected loss of PCH_PWROK. FW arms this global reset source via GBLRST_CTL.EN_PCHPWR_FLR. This register bit is in the RTC power well and is retained if the system has a functioning RTC battery during the G3 Mechanical Off state. |
| 10 | 0x0<br>RW/1C/V | **PMC Firmware Global Reset (PMC_FW):** This bit is set to '1' by hardware when a global reset is triggered by a request from PMC firmware (i.e. a write of '1' to the GBLRST_CTL.TRIG_GBL bit). |
| 9 | 0x0<br>RW/1C/V | **ME Firmware Watchdog Timer (ME_WDT):** This bit is set to '1' by hardware when a global reset is triggered by the second expiration of the ME firmware watchdog timer. |
| 8 | 0x0<br>RW/1C/V | **PMC Firmware Watchdog Timer (PMC_WDT):** This bit is set to '1' by hardware when a global reset is triggered by the second expiration of the PMC firmware watchdog timer. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C/V | **LTRESET# With Policy 1 (LTRST_POL1):** This bit is set to '1' by hardware when a global reset is triggered by an LTRESET# assertion with LT_E2STS.LT_RESET_POLICY = 1. |
| 6 | 0x0<br>RW/1C/V | **ME-Initiated Global Reset (ME_GBL):** This bit is set to '1' by hardware when a global reset is triggered by an ME FW write of 1's to both GENCTL."ME-Partition Reset" and GENCTL."ME-Initiated Host Reset With Power Cycle" in the same write cycle (this is ME FW's method of requesting a global reset). |
| 5 | 0x0<br>RW/1C/V | **CPU Thermal Trip (CPU_TRIP):** This bit is set to '1' by hardware when a global reset is triggered by a CPU thermal trip event (i.e. an assertion of the THRMTRIP# pin). |
| 4 | 0x0<br>RW/1C/V | **ME-Initiated Power Button Override (ME_PBO):** This bit is set to '1' by hardware when a global reset is triggered by an ME FW write of '1' to GENCTL."ME-Initiated Power Button Override." |
| 3 | 0x0<br>RW/1C/V | **ICH Catastrophic Temperature Event (ICH_CAT_TMP):** This bit is set to '1' by hardware when a global reset is triggered by a catastrophic temperature event from the ICH internal thermal sensor. |
| 2 | 0x0<br>RW/1C/V | **PMC SUS RAM Uncorrectable Error (PMC_UNC_ERR):** This bit is set to '1' by hardware when a global reset is triggered due to an uncorrectable parity error on a data read from one of the PMC SUS well register files. |
| 1 | 0x0<br>RW/1C/V | **Power Button Override (PB_OVR):** This bit is set to '1' by hardware when a global reset is triggered by a power button override (i.e. an assertion of the PMU_PWRBTN_N pin for 5.1 seconds). |
| 0 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |

## 58.3.14 Global Reset Causes (GBLRST_CAUSE1)—Offset 128h

This register logs causes of host partition resets.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 128h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RSVD_31_6):** Reserved. |
| 5 | 0x0<br>RW/1C/V | **ME Set Power Button Status (ME_SET_PBO_STS):** If this bit is set, the cause of the previous global reset was ME FW setting the power button override status. |
| 4 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 3 | 0x0<br>RW/1C/V | **Host SMBus Message (HSMB_MSG):** If this bit is set, the cause of the previous global reset was a global reset request received over the host SMBus interface. |
| 2 | 0x0<br>RW/1C/V | **Host Partition Reset Promotion (HOST_RST_PROM):** If this bit is set, the cause of the previous global reset was a host partition reset that was promoted to a global reset either due to ME or host policy. |
| 1 | 0x0<br>RW/1C/V | **Sx Entry Timeout (SX_ENTRY_TIMEOUT):** If this bit is set, the cause of the previous global reset was an expiration of the timer that runs during Sx entry. |
| 0 | 0x0<br>RW/1C/V | **Host Partition Reset Timeout (HOST_RESET_TIMEOUT):** If this bit is set, the cause of the previous global reset was an expiration of the timer that runs during host partition resets. |

## 58.3.15 Host Partition Reset Causes (HPR_CAUSE0)—Offset 12Ch

This register logs causes of host partition resets.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 12Ch

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Reserved (RSVD_31_19):** Reserved. |
| 18 | 0x0<br>RO/V | **IE Initiated Host Reset With Power Down (IE_HRPD):** IE initiated host reset with power down |
| 17 | 0x0<br>RO/V | **IE Initiated Host Reset With Power Cycle (IE_HRPC):** IE initiated host reset with power cycle. |
| 16 | 0x0<br>RO/V | **IE Initiated Host Rest Without Power Cycle (IE_HR):** IE initiated host reset without power cycle. |
| 15 | 0x0<br>RO | **Reserved (RSVD_15):** Reserved. |
| 14 | 0x0<br>RO/V | **LT Sleep Error (LT_SLPERR):** Sleep entry attempted while LT secrets were present in memory. |
| 13 | 0x0<br>RO/V | **Host SMBUS Host Reset With Power Cycle (HSMB_HRPC):** SMBus initiated host partition reset with power cycle. |
| 12 | 0x0<br>RO/V | **Host SMBUS Host Reset Without Power Cycle (HSMB_HR):** SMBus initiated host partition reset without power cycle. |
| 11 | 0x0<br>RO | **Reserved (RSVD_11_11):** Reserved. |
| 10 | 0x0<br>RO/V | **ME-Initiated Host Reset With Power Down (MI_HRPD):** Intel® ME initiated host reset with power down. |
| 9 | 0x0<br>RO/V | **ME-Initiated Host Reset With Power Cycle (MI_HRPC):** Intel® ME initiated host reset with power cycle. |
| 8 | 0x0<br>RO/V | **ME-Initiated Host Reset Without Power Cycle (MI_HR):** Intel® ME initiated host reset without power cycle. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 6 | 0x0<br>RO/V | **Host TCO Watchdog Timer Second Expiration (TCO_WDT):** Host TCO watchdog timer reached zero for the second time. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD_5_4):** Reserved. |
| 3 | 0x0<br>RO/V | **LTRESET (LTRST):** LTRESET message received and the LT.RESET.POLICY bit = '0'. |
| 2 | 0x0<br>RO/V | |
| 1 | 0x0<br>RO/V | **Write to CF9 (CF9_ES):** This bit will be set when Host software writes a value of 6h or Eh to the CF9h register in Host I/O Space.<br>***Note:*** The "shutdown" special cycle from the CPU will also set this bit. |
| 0 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |

## 58.3.16 modPHY Power Management Configuration Reg 1 (MODPHY_PM_CFG1)—Offset 200h

This register contains misc. fields used to configure the SoC power management behavior with respect to the modPHY.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 200h

**Default:** 0000FFFFh

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW | **ModPHY Lane S0 SUS Well Power Gating Policy [15:0] (MLS0SWPGP):** This is a bit per lane that controls SUS Well Power Gating for a ModPHY lane to be used for S0 and S0ix use models as described above in HAS.<br>• Bit 0: Corresponds to ModPHY Lane 0<br>• Bit 1: Corresponds to ModPHY Lane 1<br>• Bit 2: Corresponds to ModPHY Lane 2<br>:<br>:<br>Bit 15: Corresponds to ModPHY Lane 15<br>For each lane:<br>• 0: Lane power gating not permitted in S0.<br>• 1: Lane power gating is permitted in S0.<br>Note that strap information/PCIe Lane reversal information will be factored by BIOS when programming these bits in the configuration registers.<br>Note that it is illegal SW programming to have a bit location to be 1 in this field and the corresponding bit position to be 0 in MLSXSWPGP |
| 15:0 | 0xffff<br>RW | • **ModPHY Lane Sx SUS Well Power Gating Policy [15:0] (MLSXSWPGP):** This is a bit per lane that controls SUS Well Power Gating for a ModPHY lane when system is in Sx.<br>• Bit 0: Corresponds to ModPHY Lane 0<br>• Bit 1: Corresponds to ModPHY Lane 1<br>• Bit 2: Corresponds to ModPHY Lane 2<br>:<br>:<br>Bit 15: Corresponds to ModPHY Lane 15<br>For each lane:<br>• 0: Lane power gating not permitted in Sx.<br>• 1: Lane power gating is permitted in Sx.<br>Note that strap information/PCIe Lane reversal information will be factored by BIOS when programming these bits in the configuration registers.<br>For ease of PMC implementation, this field will be used to manage Sx policies even in S0. In other words, the earlier restriction that BIOS does not have to program this field if MLSPDDGE is 1 does not apply any more.<br>BIOS shall set this field appropriately for all cases. |

## 58.3.17 modPHY Power Management Configuration Reg 2 (MODPHY_PM_CFG2)—Offset 204h

This register contains misc. fields used to configure the SoC power management behavior with respect to the modPHY.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 204h

**Default:** 05000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Gen2PLL Request Control (G2PLLREQCTL):** This bit controls how PMC should treat Gen2PLL power request for ModPHY power gating flows.<br>• 0 - PMC should treat Gen2PLL request as non restore power request<br>• 1 - PMC should treat Gen2PLL request as restore power request |
| 30 | 0x0<br>RW | **ModPHY Lane SUS Power Domain Dynamic Gating Enable (MLSPDDGE):**<br>When this bit is set to 1, ModPHY Lane SUS Well Dynamic Gating is enabled.<br>ModPHY lane SPD is managed using the variables defined by the PMC.<br>When this bit is 0, ModPHY Lane SUS Well Gating can still be done at a more coarse level using MLSXSWPGP and MLS0SWPGP fields. |
| 29 | 0x0<br>RW | **Enable ModPHY FET Control (EMFC):** This bit enables PMC dynamic control of ModPHY external FET.<br>When this bit is 0 and MLSPDDGE is 1, PMC goes through all of the ModPHY power gating flows except that the external FET is not turned off.<br>This bit is being provided primarily to prevent External FET gating during EXI debug |
| 28:24 | 0x5<br>RW/L | **External FET Ramp Time (EFRT):** This field defines the ramp time of ModPHY FET.<br>• 00000b: 00us<br>• 00001b: 20us<br>• 00010b: 40us<br>• ...<br>• 00101b: 100us<br>• ...<br>• 11111b: 620us<br>This bit is locked while PM_SYNC_MISC_CFG.PM_SYNC_LOCK = '1' |
| 23:20 | 0x0<br>RO | **Reserved (RSVD_23_20):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 19:0 | 0x0<br>RW | **ASL Over-rides [19:0] (ASLOR):** This field provides ASL code to take over SPD power gating control.<br>If ASL code sets a bit corresponding to a controller, it implies that PMC shall ignore any other inputs from that controller and assume that the controller is ready for SPD power gating.<br>0 in a bit position: Use HW interfaces for the controller to manage SPD gating/wake-up.<br>1 in a bit position: Ignore HW interfaces for the controller, rely only on MSPDRTReq field thats managed by ASL code.<br>• Bit 0: Corresponds to PCIe Controller A, Function 0<br>• Bit 1: Corresponds to PCIe Controller A, Function 1<br>• Bit 2: Corresponds to PCIe Controller A, Function 2<br>• Bit 3: Corresponds to PCIe Controller A, Function 3<br>• Bit 4: Corresponds to PCIe Controller B, Function 0<br>• Bit 5: Corresponds to PCIe Controller B, Function 1<br>• Bit 6: Corresponds to PCIe Controller B, Function 2<br>• Bit 7: Corresponds to PCIe Controller B, Function 3<br>• Bit 8: Corresponds to PCIe Controller C, Function 0<br>• Bit 9: Corresponds to PCIe Controller C, Function 1<br>• Bit 10: Corresponds to PCIe Controller C, Function 2<br>• Bit 11: Corresponds to PCIe Controller C, Function 3<br>• Bit 12: Corresponds to SATA Controller<br>• Bit 13: Corresponds to Gbe Controller<br>• Bit 14: Corresponds to xHCI Controller<br>• Bit 15: Corresponds to xDCI Controller<br>• Bit 16: Corresponds to UFS Controller<br>• Bits[19:17]: Reserved<br>This field is going to be used in conjunction with MSPDRTReq and MSPDRTRAck fields above. If ASL code intends to over-ride HW decisions, it will set the corresponding bit for a controller/function to 1 in ASLOR and use MSPDRTReq bits to power-up/power-down SPD. |

## 58.3.18 modPHY Power Management Configuration Reg 3 (MODPHY_PM_CFG3)—Offset 208h

This register contains misc. fields used to configure the SoC power management behavior with respect to the modPHY.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 208h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **Reserved (RSVD_31_20):** Reserved. |
| 19:0 | 0x0<br>RW | **Controller SPD RTD3 Request [19:0] (MSPDRTREQ):** This field represents ASL code trigger request for ModPHY SPD gating.<br>If this bit is set (to 1) for a controller, it implies that ASL code provides consent for SPD to be gated for the corresponding controllers lanes.<br>Note that this bit could also be more statically used by BIOS to set this to 1 for a controller where SPD will only be managed through other interfaces implying ASL code does not exist for a controller.<br>This is not a POR mode of operation if a function is enabled ASL code will exist for all controllers that are enabled. However, the controllers that are not enabled (Function Disabled), this field will be statically set by BIOS to activate ASL component in SPD gating equations.<br>• Bit 0: Corresponds to PCIe Controller A, Function 0<br>• Bit 1: Corresponds to PCIe Controller A, Function 1<br>• Bit 2: Corresponds to PCIe Controller A, Function 2<br>• Bit 3: Corresponds to PCIe Controller A, Function 3<br>• Bit 4: Corresponds to PCIe Controller B, Function 0<br>• Bit 5: Corresponds to PCIe Controller B, Function 1<br>• Bit 6: Corresponds to PCIe Controller B, Function 2<br>• Bit 7: Corresponds to PCIe Controller B, Function 3<br>• Bit 8: Corresponds to PCIe Controller C, Function 0<br>• Bit 9: Corresponds to PCIe Controller C, Function 1<br>• Bit 10: Corresponds to PCIe Controller C, Function 2<br>• Bit 11: Corresponds to PCIe Controller C, Function 3<br>• Bit 12: Corresponds to SATA Controller<br>• Bit 13: Corresponds to Gbe Controller<br>• Bit 14: Corresponds to xHCI Controller<br>• Bit 15: Corresponds to xDCI Controller<br>• Bit 16: Corresponds to UFS Controller<br>• Bits[19:17]: Reserved |

## 58.3.19 modPHY Power Management Configuration Reg 4 (MODPHY_PM_CFG4)—Offset 20Ch

This register contains misc. fields used to configure the SoC power management behavior with respect to the modPHY.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 20Ch

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RO | **Reserved (RSVD_31_20):** Reserved. |
| 19:0 | 0x0<br>RO/V | **Controller SPD RTD3 Request Acknowledge [19:0] (MSPDRTRACK):** This field represents the acknowledge for ASL code trigger request for ModPHY SPD gating.<br>PMC sets a bit in this field to 1 to acknowledge that it has registered the corresponding MSPDRTREQ. Note that the action of setting this bit to 1 is immediate no other gating conditions are involved in this. Actual SPD shutdown may happen later once other power gating conditions have been satisfied as well.<br>PMC clears a bit to 0 in this field once the corresponding MSPTDRTREQ is cleared by the ASL code and SPD state has been fully restored.<br>• Bit 0: Corresponds to PCIe Controller A, Function 0<br>• Bit 1: Corresponds to PCIe Controller A, Function 1<br>• Bit 2: Corresponds to PCIe Controller A, Function 2<br>• Bit 3: Corresponds to PCIe Controller A, Function 3<br>• Bit 4: Corresponds to PCIe Controller B, Function 0<br>• Bit 5: Corresponds to PCIe Controller B, Function 1<br>• Bit 6: Corresponds to PCIe Controller B, Function 2<br>• Bit 7: Corresponds to PCIe Controller B, Function 3<br>• Bit 8: Corresponds to PCIe Controller C, Function 0<br>• Bit 9: Corresponds to PCIe Controller C, Function 1<br>• Bit 10: Corresponds to PCIe Controller C, Function 2<br>• Bit 11: Corresponds to PCIe Controller C, Function 3<br>• Bit 12: Corresponds to SATA Controller<br>• Bit 13: Corresponds to Gbe Controller<br>• Bit 14: Corresponds to xHCI Controller<br>• Bit 15: Corresponds to xDCI Controller<br>• Bit 16: Corresponds to UFS Controller<br>• Bits[19:17]: Reserved |

## 58.3.20 IE_STS (IE_STS)—Offset 3F0h

This register contains misc fields used to record events pertaining to the Innovation Engine (IE) status. This is shadowed from PMC XRAM IEPMMISC register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 3F0h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO/V | **Shadowed IEPMMISC[31:24] (SHAD_IEPMMISC_31_24):** Shadowed from PMC_XRAM_IEPMMISC |
| 23 | 0x0<br>RO/V | **Shadowed IEPMMISC[23] (SHAD_IEPMMISC):** Shadowed IEPMMISC[23] |
| 22 | 0x0<br>RO/V | **Shadowed Bios Authentication Result (SHAD_IE_BIOS_AUTH):** Written by IE OEM FW after it attempts to authenticate BIOS image.<br>• 0 = BIOS authentication failed<br>• 1 = BIOS authentication succeeded<br>***Note:*** This field is made available to OEM FW for its use. |
| 21 | 0x0<br>RO/V | **Shadowed FW Authentication Result (SHAD_IE_FW_AUTH):**<br>• 0 = IE FW authentication failed<br>• 1 = IE FW authentication succeeded |
| 20 | 0x0<br>RO/V | **Shadowed IE Hash Valid Fuse (SHAD_IE_HASHVALID_FUSE):** Written by IE ROM. Shadows the IE_HASHVALID_FUSE (OEM IFP) pulled by IE.<br>• 0 = use debug Hash<br>• 1 = use OEM Hash<br>FYI to ME FW and BIOS. |
| 19 | 0x0<br>RO/V | **Shadowed IE Locked (SHAD_IE_LOCKED):** Written by IE ROM. Shadows the IE_EOM_FUSE (OEM IFP) pulled by IE.<br>• 0 = OEM in manufacturing (default)<br>• 1 = OEM End-of-Manufacturing<br>FYI to ME FW and BIOS. |
| 18 | 0x0<br>RO/V | **Shadowed IE Verified Policy Fuse (SHAD_IE_VBPOLICY_FUSE):** Shadowed from PCI MMR_IE_STS |
| 17 | 0x0<br>RO/V | **Shadowed IE Disable Strap (SHAD_IE_DISABLE_STRAP):** Written by IE ROM. Shadows the IE_DISABLE_STRAP (soft strap) pulled by IE. FYI to ME FW and BIOS. |
| 16 | 0x0<br>RO/V | **Shadowed IE Policy Done (SHAD_IE_POLICY_DONE):** Set by IE ROM after it has set IE_DIS_HOST_WAK_SUS. This bit is a handshake from IE ROM to PMC FW.<br>• 1) PMC FW should treat IE_POLICY_DONE (IE) similar to PATCH_DONE (Intel® ME): both are used as handshakes; the reset sequence will pause until set. Also both conditions have (chicken-out) PMC timers that, upon expiration, will cause reset to proceed regardless of the handshake.<br>• 2) PMC FW should use IE_POLICY_DONE handshake to comprehend IE_DIS_HOST_WAK_SUS, even if S0_COMMIT was already asserted.<br>IE ROM must set IE_DIS_HOST_WAK_SUS (to prevent boot if VB fails).<br>***Note:*** See description of PPS.S0_COMMIT and GENCTL.DIS_HOST_WAK_SUS for more details. |
| 15:9 | 0x0<br>RO/V | **Shadowed IE_STS[15:9] (SHAD_IE_STS_15_9):** Shadowed from PMC XRAM IEPMMISC |
| 8 | 0x0<br>RO/V | **Shadowed IE_CIOFF_STS (SHAD_IE_CIOFF_STS):** Shadows IE_CIOFF_STS bit in XRAM space, which is controlled by PMC FW and set when CI0-)CIoff is completed.<br>BIOS shall poll this bit, after requesting CI0-)Cioff (indicated by IE_CIOFF_REQ). Once set BIOS should re-enumerate to remove IE devices and handoff to O/S.<br>O/S-discovered devices should be persistent. |
| 7:2 | 0x0<br>RO/V | **Shadowed IE_STS[7:2] (SHAD_IE_STS_7_2):** Shadowed from PMC XRAM IEPMMISC |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1:0 | 0x0<br>RO/V | **Shadowed IE_STS[1:0] (SHAD_IE_STS_1_0):** Shadowed from PMC XRAM IEPMMISC |

## 58.3.21 ME_STS (ME_STS)—Offset 3F4h

This register contains misc. fields used to record events pertaining to the Intel® Management Engine status. This is shadowed from PMC XRAM MEPMMISC register.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 3F4h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD_31_16):** Reserved. |
| 15:9 | 0x0<br>RO/V | **Shadowed ME_STS[15:9] (SHAD_ME_STS_15_9):** Shadowed from PMC_XRAM MEPMMISC[15:9] |
| 8 | 0x0<br>RO/V | **Shadowed ME_CMOFF_STS (SHAD_ME_CMOFF_STS):** Shadows ME_CMOFF_STS in XRAM space, which is controlled by PMC FW and set when CM0-)CMoff is completed.<br>BIOS shall poll this bit, after requesting CM0-)CMoff (indicated by IE_CMOFF_REQ). Once set BIOS should re-enumerate to remove ME devices and handoff to O/S.<br>Why? O/S-discovered devices should be persistent.<br>Shadowed from PMC XRAM MEPMMISC[8] |
| 7:0 | 0x0<br>RO/V | **Shadowed ME_STS[7:0] (SHAD_ME_STS_7_0):** Shadowed from PMC_XRAM MEPMMISC[7:0] |

## 58.3.22 Static PG Related Function Disable Register 1 (ST_PG_FDIS_PMC_1)—Offset 620h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [PWRMBASE] + 620h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Static Function Disable Lock (ST_FDIS_LK):** Lock control for all ST_PG_FDIS_PMC_* and NST_PG_FDIS_* registers (also self-locks when written to 1). |
| 30:13 | 0x0<br>RO | **Reserved (RSVD_30_13):** Reserved. |
| 12 | 0x0<br>RW/L | **EXI Function Disable (PMC Version) (EXI_FDIS_PMC):** BIOS is required to set this bit when EXI function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 11 | 0x0<br>RW/L | **NPK Function Disable (PMC Version) (NPK_FDIS_PMC):** BIOS is required to set this bit when NPK function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 10 | 0x0<br>RW/L | **IE Function Disable (PMC Version) (IE_FDIS_PMC):** BIOS is required to set this bit when IE function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 9 | 0x0<br>RW/L | **ME Function Disable (PMC Version) (ME_FDIS_PMC):** BIOS is required to set this bit when ME function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 8 | 0x0<br>RW/L | **XHCI Function Disable (PMC Version) (XHCI_FDIS_PMC):** BIOS is required to set this bit when XHCI function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 7 | 0x0<br>RW/L | **eMMC (SCC) Function Disable (PMC Version) (EMMC_FDIS_PMC):** BIOS is required to set this bit when eMMC function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 6 | 0x0<br>RW/L | **nCPM Function Disable (PMC Version) (NCPM_FDIS_PMC):** BIOS is required to set this bit when Intel® QuickAssist Technology function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 5 | 0x0<br>RW/L | **PCIE1 Function Disable (PMC Version) (PCIE1_FDIS_PMC):** BIOS is required to set this bit when PCIE Ctrl 1 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 4 | 0x0<br>RW/L | **PCIE0 Function Disable (PMC Version) (PCIE0_FDIS_PMC):** BIOS is required to set this bit when PCIE Ctrl 0 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 3 | 0x0<br>RW/L | **SATA1 Function Disable (PMC Version) (SATA1_FDIS_PMC):** BIOS is required to set this bit when SATA Ctrl 1 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 2 | 0x0<br>RW/L | **SATA0 Function Disable (PMC Version) (SATA0_FDIS_PMC):** BIOS is required to set this bit when SATA Ctrl 0 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 1 | 0x0<br>RW/L | **GBE1 Function Disable (PMC Version) (GBE1_FDIS_PMC):** BIOS is required to set this bit when GBE Ctrl 1 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 0 | 0x0<br>RW/L | **GBE0 Function Disable (PMC Version) (GBE0_FDIS_PMC):** BIOS is required to set this bit when GBE Ctrl 0 function is configured to be function disabled. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |

## 58.3.23 Non-Static PG Related Function Disable Register 1 (NST_PG_FDIS_1)—Offset 628h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 628h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3:0 | 0x0<br>RW | **LAN port disable control bits (LAN_PORT_DISABLE):** Host disable of the LAN ports, LAN_PORT_DISABLE[3:0] = Odem1[1:0] + Odem0[1:0] |

## 58.3.24 Non-Static PG Fuse Disable Read 1 Register (N_STPG_FUSE_SS_DIS_RD_1)—Offset 640h

This register contains fuse disable based information - which is not used for static power gating (this information is provided only for host/BIOS read access).

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 640h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:4 | 0x0<br>RO | **Reserved (RSVD_31_4):** Reserved. |
| 3:0 | 0x0<br>RO/V | **LAN port Disable status (LAN_PORT_DISABLE_STATUS):** Result of all fuses and straps and control bits of the LAN port disable, LAN_PORT_DISABLE_STATUS[3:0] = Odem1[1:0] + Odem0[1:0] |

## 58.3.25 Static PG Fuse and Soft Strap Disable Read Register 2 (STPG_FUSE_SS_DIS_RD_2)—Offset 644h

This register contains fuse and soft strap disable based information - all of which is used for static power-gating (for example, if a bit is set in this register for a particular IP, PMC keeps that IP in a power-gated state - except as noted below.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [PWRMBASE] + 644h

**Default:** 00000000h

**PWRMBASE Type:** PCI Configuration Register (Size: 32 bits)
**PWRMBASE Reference:** [B:0, D:31, F:2] + 48h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD_31_16):** Reserved. |
| 15 | 0x0<br>RO/V | **Intel® Management Engine (Intel® ME) Fuse or Soft Strap Disable (ME_FUSE_SS_DIS):** RO bit indicating if the Intel® Management Engine function is disabled through fuse/soft strap. |
| 14 | 0x0<br>RO/V | **Innovation Engine Fuse or Soft Strap Disable (IE_FUSE_SS_DIS):** RO bit indicating if Innovation Engine function is disabled through fuse/soft strap. |
| 13 | 0x0<br>RO/V | **USB2 Fuse or Soft Strap Disable (USB2_FUSE_SS_DIS):** RO bit indicating if USB2 function is disabled through fuse/soft strap. |
| 12 | 0x0<br>RO/V | **LPC Fuse or Soft Strap Disable (LPC_FUSE_SS_DIS):** RO bit indicating if LPC function is disabled through fuse/soft strap. |
| 11 | 0x0<br>RO/V | **ESPISPI Fuse or Soft Strap Disable (ESPISPI_FUSE_SS_DIS):** RO bit indicating if ESPISPI function is disabled through fuse/soft strap. |
| 10 | 0x0<br>RO/V | **NPK Fuse or Soft Strap Disable (NPK_FUSE_SS_DIS):** RO bit indicating if NPK function is disabled through fuse/soft strap. |
| 9 | 0x0<br>RO | **Reserved (RSVD_9):** Reserved. |
| 8 | 0x0<br>RO/V | **eMMC Fuse or Soft Strap Disable (EMMC_FUSE_SS_DIS):** RO bit indicating if eMMC (SCC) function is disabled through fuse/soft strap. |
| 7 | 0x0<br>RO/V | **XHCI Fuse or Soft Strap Disable (XHCI_FUSE_SS_DIS):** RO bit indicating if XHCI function is disabled through fuse/soft strap. |
| 6 | 0x0<br>RO/V | **nCPM Fuse or Soft Strap Disable (NCPM_FUSE_SS_DIS):** RO bit indicating if Intel® QuickAssist Technology function is disabled through fuse/soft strap. |
| 5 | 0x0<br>RO/V | **PCIE Ctrl 1 Fuse or Soft Strap Disable (PCIE1_FUSE_SS_DIS):** RO bit indicating if PCIE Ctrl 1 function is disabled through fuse or Soft strap. |
| 4 | 0x0<br>RO/V | **PCIE Ctrl 0 Fuse or Soft Strap Disable (PCIE0_FUSE_SS_DIS):** RO bit indicating if PCIE Ctrl 0 function is disabled through fuse or Soft strap. |
| 3 | 0x0<br>RO/V | **SATA Ctrl 1 Fuse or Soft Strap Disable (SATA1_FUSE_SS_DIS):** RO bit indicating if SATA Ctrl 1 function is disabled through fuse or Soft strap. |
| 2 | 0x0<br>RO/V | **SATA Ctrl 0 Fuse or Soft Strap Disable (SATA0_FUSE_SS_DIS):** RO bit indicating if SATA Ctrl 0 function is disabled through fuse or Soft strap. |
| 1 | 0x0<br>RO/V | **GBE Ctrl 1 Fuse or Soft Strap Disable (GBE1_FUSE_SS_DIS):** RO bit indicating if GBE Ctrl 1 function is disabled through fuse or Soft strap. |
| 0 | 0x0<br>RO/V | **GBE Ctrl 0 Fuse or Soft Strap Disable (GBE0_FUSE_SS_DIS):** RO bit indicating if GBE Ctrl 0 function is disabled through fuse or Soft strap. |

## 58.4 Registers in I/O Space—ABASE

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) ABASE described in Section 58.2.2, "ACPI Base Address (ABASE)—Offset 40h" on page 2689.

### 58.4.1 Power Management 1 Enables and Status (PM1_EN_STS)—Offset 0h

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 0h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD_31):** Reserved. |
| 30 | 0x0<br>RW | **PCI Express Wake Disable (PCIEXP_WAKE_DIS):** This bit disables the inputs to the PCIEXP_WAKE_STS bit in the PM1 Status register from waking the system.<br>Modification of this bit has no impact on the value of the PCIEXP_WAKE_STS bit. |
| 29:27 | 0x0<br>RO | **Reserved (RSVD_29_27):** Reserved. |
| 26 | 0x0<br>RW/V | **RTC Alarm Enable (RTC_EN):** This is the RTC alarm enable bit. It works in conjunction with the SCI_EN bit. When RTC_STS is set, one of the following happens:<br>• When RTC_EN=0: No SMI# or SCI. If system was in S1-S5, no wake event occurs.<br>• When RTC_EN=1 and SCI_EN = 1: SCI. If system was in S1-S5, then a wake event occurs before the SCI.<br>• When RTC_EN=1 and SCI_EN = 0: SMI#. If system was in S1-S5, then a wake event occurs before the SMI#.<br>This register bit is circuitry in the SoC RTC power well in order to allow an RTC event to wake after a power failure. It is reset only when the active-low SRTCRST_N input signal pin is asserted. |
| 25 | 0x0<br>RO | **Reserved (RSVD_25):** Reserved. |
| 24 | 0x0<br>RW/V | **Power Button Enable (PWRBTN_EN):** This bit is the power button enable. It works in conjunction with the SCI_EN bit. When PWRBTN_STS is set, one of the following happens:<br>• When PWRBTN_EN=0: No SMI# or SCI.<br>• When PWRBTN_EN=1 and SCI_EN = 1: SCI.<br>• When PWRBTN_EN=1 and SCI_EN = 0: SMI#.<br>***Note:*** PWRBTN_EN has no effect on the PWRBTN_STS bit being set by the assertion of the power button. The Power Button is always enabled as a Wake event. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_23_22):** Reserved. |
| 21 | 0x0<br>RW | **Global Enable (GBL_EN):** The global enable bit. When both the GBL_EN and the GBL_STS are set, the SoC generates an SCI. |
| 20:17 | 0x0<br>RO | **Reserved (RSVD_20_17):** Reserved. |
| 16 | 0x0<br>RW | **Timer Overflow Interrupt Enable (TMROF_EN):** This is the timer overflow interrupt enable bit. It works in conjunction with the SCI_EN bit. When TMROF_STS is set, one of the following happens:<br>• When TMROF_EN=0: No SMI# or SCI.<br>• When TMROF_EN=1 and SCI_EN = 1: SCI.<br>When TMROF_EN=1 and SCI_EN = 0: SMI#. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C/V | **Wake Status (WAK_STS):** This bit is set when the system is in one of the Sleep states (via the SLP_EN bit) and an enabled SoC Wake event occurs. Upon setting this bit, the SoC will transition the system to the ON state.<br>This bit can only be set by hardware and can only be cleared by writing a one to this bit position.<br>This bit is not affected by hard resets caused by a CF9 write, but is reset by the SoC-internal pri_pwrgood_rst_b signal. |
| 14 | 0x0<br>RW/1C/V | **PCI Express Wake Status (PCIEXP_WAKE_STS):** This bit is set by hardware to indicate that the system woke due to a PCI Express wakeup event.<br>This event can be caused by the PCI Express WAKE# pin being active, or one or more of the PCI Express ports being in beacon state, or receipt of a PCI Express PME message at root port. |
| 13:12 | 0x0<br>RO | **Reserved (RSVD_13_12):** Reserved. |
| 11 | 0x0<br>RW/1C/V | **Power Button Override (PWRBTNOR_STS):** This bit is set when any of these occur:<br>• A Power Button Override Event occurs (i.e., the power button is pressed for at least four consecutive seconds)<br>• The corresponding bit is received in the SMBus slave message<br>• The Intel® ME-Initiated "Power Button Override" bit is set<br>• The Intel® ME-Initiated "Host Reset with Power Down" is set<br>• An internal thermal sensor catastrophic condition occurs<br>These events cause an unconditional transition to the S5 state. |
| 10 | 0x0<br>RW/1C/V | **RTC Status (RTC_STS):** This bit is set when the RTC generates an alarm (assertion of the IRQ8# signal), and is not affected by any other enable bit. See the RTC_EN bit in this register for the effect when RTC_STS goes active. |
| 9 | 0x0<br>RO | **Reserved (RSVD_9):** Reserved. |
| 8 | 0x0<br>RW/1C/V | **Power Button Status (PWRBTN_STS):** This bit is set when the PMU_PWRBTN_N input signal pin is asserted (low), independent of any other enable bit. See the PWRBTN_EN bit in this register for the effect when PWRBTN_STS goes active.<br>PWRBTN_STS is always a wake event. This bit is only set by hardware and can be cleared by software writing a one to this bit position. This bit is not affected by hard resets caused by a CF9 write, but is reset by the SoC internal pwrgood_rst_b. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_7_6):** Reserved. |
| 5 | 0x0<br>RW/1C/V | **GBL Status (GBL_STS):** This bit is set when an SCI is generated due to the BIOS wanting the attention of the SCI handler. BIOS has a corresponding bit, BIOS_RLS, which will cause an SCI and set this bit. The SCI handler should then clear this bit by writing a 1 to it.<br>This bit will not cause wake events or SMI#.<br>This bit is not affected by SCI_EN. |
| 4 | 0x0<br>RW/1C/V | **Bus Master Status (BM_STS):** This bit is set to 1 by the SoC when a SoC-visible bus master requests access to memory or the BM_BUSY# signal is active. This bit is cleared by the CPU writing a 1 to this bit position. This bit will not cause a wake event, SCI, or SMI. |
| 3:1 | 0x0<br>RO | **Reserved (RSVD_3_1):** Reserved. |
| 0 | 0x0<br>RW/1C/V | **Timer Overflow Status (TMROF_STS):** This is the timer overflow status bit. |

## 58.4.2 Power Management 1 Control —Offset 4h

Usage: ACPI or Legacy. Not lockable. Bits 0-9, 13-31 = Primary Power Well. Bits 10-12 = RTC Power Well.

**Type:** I/O Register (Size: 32 bits)

**BAR and Offset:** [ABASE] + 4h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved (RSVD_31_14):** Reserved. |
| 13 | 0x0<br>WO | **Sleep Enable (SLP_EN):** This is a write-only bit and reads to it always return a zero.<br>Setting this bit causes the system to sequence into the Sleep state defined by the SLP_TYP field. |
| 12:10 | 0x0<br>RW | **Sleep Type (SLP_TYP):** This 3-bit field defines the type of Sleep the system should enter when the SLP_EN bit is set to 1. These bits are in the RTC power well and are cleared when the SoC input signal pin SRTCRST_N is active.<br>• 000: Mode = ON, Typical Mapping = S0<br>• 001: Reserved<br>• 010: Reserved<br>• 011: Reserved<br>• 100: Reserved<br>• 101: Reserved<br>• 110: Mode = Suspend-To-Disk, Typical Mapping = S4<br>• 111: Mode = Soft Off, Typical Mapping = S5 |
| 9:3 | 0x0<br>RO | **Reserved (RSVD_9_3):** Reserved. |
| 2 | 0x0<br>WO | **GBL_RLS:** The Global Release bit is used by the ACPI software which writes a '1' to raise an event to the BIOS software.<br>This bit always reads as 0. |
| 1 | 0x0<br>RW | **BM_RLD:** This bit is treated as a scratchpad bit. |
| 0 | 0x0<br>RW | **SCI Enable (SCI_EN):** Selects the SCI interrupt or the SMI# for various events.<br>• 0 = These events will generate an SMI#.<br>• 1 = These events will generate an SCI. |

## 58.4.3 Power Management 1 Timer (PM1_TMR)—Offset 8h

Usage: ACPI. Not lockable. Primary Power Well.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 8h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD_31_24):** Reserved. Will always read 0. |
| 23:0 | 0x0<br>RO/V | **Timer Value (TMR_VAL):** This read-only field returns the running count of the PM timer.<br>This counter runs off a 3.579545 MHz clock (derived from 14.31818 MHz divided by 4).<br>It is reset (to 0) during a PCI reset, and then continues counting as long as the system is in the S0 state.<br>Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit is set. The High-to-Low transition will occur every 2.3435 seconds. Writes to this register have no effect. |

## 58.4.4 SMI Control and Enable (SMI_EN)—Offset 30h

Usage: ACPI or Legacy. Not lockable. Primary Power Well.
This register is symmetrical to the SMI Status Register.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 30h

**Default:** 00000002h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **XHCI SMI Enable (XHCI_SMI_EN):** Software sets this bit to enable XHCI SMI events. |
| 30 | 0x0<br>RW | **ME SMI Enable (ME_SMI_EN):** Software sets this bit to enable Intel® ME SMI# events. |
| 29 | 0x0<br>RW | **LPSS SMI Enable (LPSS_SMI_EN):** Software sets this bit to enable LPSS SMII events. |
| 28 | 0x0<br>RW/L | **Reserved:** Reserved. |
| 27 | 0x0<br>RW/1S | **GPIO Unlock SMI Enable (GPIO_UNLOCK_SMI_EN):** Setting this bit will cause the SoC to generate an SMI# when the GPIO_UNLOCK_SMI_STS bit is set in the SMI_STS register.<br>Once written to '1', this bit can only be cleared by host_prim_rst_b. |
| 26 | 0x0<br>RO | **Reserved (RSVD_26):** Reserved. |
| 25 | 0x0<br>RW | **SCC SMI Enable (SCC_SMI_EN):** Software sets this bit to enable SCC (eMMC) SMI events |
| 24 | 0x0<br>RO | **Reserved (RSVD_24):** Reserved. |
| 23 | 0x0<br>RW | **IE SMI Enable (IE_SMI_EN):** Software sets this bit to enable Innovation Engine (IE) SMI# events. |
| 22:19 | 0x0<br>RO | **Reserved (Rsvd_22_19):** Reserved. |
| 18 | 0x0<br>RW | **IEH (RAS Error Collector) SMI Enable (IEH_SMI_EN):** Software sets this bit to enable Internal Error Handler (IEH) SMI# events. |
| 17 | 0x0<br>RW | **Legacy USB 2 Enable (LEGACY_USB2_EN):** Enables legacy USB 2.0 logic to cause SMI#. |
| 16:15 | 0x0<br>RO | **Reserved (RSVD_16_15):** Reserved. |
| 14 | 0x0<br>RW | **Periodic Enable (PERIODIC_EN):** Setting this bit will cause the SoC to generate an SMI# when the PERIODIC_STS bit is set in the SMI_STS register. |
| 13 | 0x0<br>RW/L | **TCO Enable (TCO_EN):**<br>• 0 = Disables TCO logic generating an SMI#<br>• 1 = Enables the TCO logic to generate SMI#<br>If the NMI2SMI_EN bit is set, then SMI's that are caused by NMI's (i.e. rerouted) will not be gated by the TCO_EN bit. Even if the TCO_EN bit is 0, the NMI's will still be routed to cause the SMI#.<br>This bit can not be written once the TCO_LOCK bit (at offset 08h of TCO I/O Space) is set. This prevents unauthorized software from disabling the generation of TCO-based SMI's |
| 12 | 0x0<br>RO | **Reserved (RSVD_12):** Reserved. |
| 11 | 0x0<br>RW | **MCSMI Enable (MCSMI_EN):** Software sets this bit to 1 to enables the SoC to trap access to the microcontroller range (62h or 66h). |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RO | **Reserved (RSVD_10_8):** Reserved. |
| 7 | 0x0<br>WO | **BIOS_RLS:** Enables the generation of an SCI interrupt for ACPI software when a '1' is written to this bit position by BIOS software. This bit always reads a zero.<br>GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place. |
| 6 | 0x0<br>RW | **Software SMI Timer Enable (SWSMI_TMR_EN):** Software sets this bit to a '1' to start the Software SMI# Timer.<br>When the timer expires (depending on the SWSMI_RATE_SEL bits), it will generate an SMI# and set the SWSMI_TMR_STS bit. The SWSMI_TMR_EN bit will remain at 1 until software sets it back to 0.<br>Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated. The default for this bit is 0. |
| 5 | 0x0<br>RW | **APMC Enable (APMC_EN):** If set, this enables writes to the APM_CNT register to cause an SMI# |
| 4 | 0x0<br>RW | **SMI On Sleep Enable (SMI_ON_SLP_EN):** If this bit is set, the SoC will generate an SMI# when a write access attempts to set the SLP_EN bit (in the PM1_CNT register).<br>Furthermore, the SoC will not put the system to a sleep state.<br>It is expected that the SMI# handler will turn off the SMI_ON_SLP_EN bit before actually setting the SLP_EN bit. |
| 3 | 0x0<br>RW | **Legacy USB Enable (LEGACY_USB_EN):** Enables legacy USB circuit to cause SMI#. |
| 2 | 0x0<br>RW | **BIOS Enable (BIOS_EN):** Enables the generation of SMI# when ACPI software writes a '1' to the GBL_RLS bit.<br>***Note:*** If the BIOS_STS bit, which gets set when software writes a 1 to GBL_RLS bit, is already a 1 at the time that BIOS_EN becomes 1, an SMI# will be generated when BIOS_EN gets set. |
| 1 | 0x1<br>RW/1S/V | **End of SMI (EOS):** This bit controls the arbitration of the SMI signal to the processor.<br>This bit must be set in order for the SoC to assert SMI# low to the processor after SMI# has been asserted previously. |
| 0 | 0x0<br>RW/L | **Global SMI Enable (GBL_SMI_EN):** When set, this bit enables the generation of SMIs in the system upon any enabled SMI event.<br>This bit is reset by a PCI reset event. If this bit is not set, no SMI# will be generated.<br>When the SMI_LOCK bit is set, this bit cannot be changed. |

## 58.4.5 SMI Status Register (SMI_STS)—Offset 34h

Usage: ACPI or Legacy. Not lockable. Primary Power Well.
If the corresponding _EN bit is set when the _STS bit is set, the SoC will cause an SMI# (except bits 8-10, which don't cause SMI#).

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 34h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **XHCI SMI Status (XHCI_SMI_STS):** This bit will be set when any USB 3.0 (XHCI) Host Controller is requesting an SMI. |
| 30 | 0x0<br>RO/V | **ME SMI Status (ME_SMI_STS):** This bit will be set when the Intel® ME is requesting an SMI#. |
| 29 | 0x0<br>RW/1C/V | **LPSS SMI Status (LPSS_SMI_STS):** This bit gets set when LPSS agent is requesting SMI #.<br>This bit is set by hardware and cleared by software writing a 1 to this bit position |
| 28 | 0x0<br>RO/V | **Reserved:** Reserved. |
| 27 | 0x0<br>RW/1C/V | **GPIO Unlock SMI Status (GPIO_UNLOCK_SMI_STS):** This bit will be set of the GPIO registers lock-down logic is requesting an SMI#. Writing a '1' to this bit position clears this bit to '0'. |
| 26 | 0x0<br>RO/V | **SPI_SMI Status (SPI_SMI_STS):** This bit will be set when the SPI logic is requesting an SMI#.<br>This bit is read only because the sticky status and enable bits associated with this function are located in the SPI registers. |
| 25 | 0x0<br>RW/1C/V | **SCC SMI Status (SCC_SMI_STS):** This bit gets set when SCC agent is requesting SMI#.<br>This bit is set by hardware and cleared by software writing a 1 to this bit position. |
| 24 | 0x0<br>RO | **Reserved (RSVD_24):** Reserved. |
| 23 | 0x0<br>RO/V | **IE SMI Status (IE_SMI_STS):** This bit will be set when the Innovation Engine (IE) is requesting an SMI#. |
| 22 | 0x0<br>RO/V | **INTERNAL_TT Status (INTERNAL_TT_STS):** This bit will be set if the internal thermal throttle mechanism has changed state.<br>This bit is read only because the sticky status and enable bit associated with this function is in the other unit. |
| 21 | 0x0<br>RO/V | **Monitor Status (MONITOR_STS):** This bit will be set if the Trap/SMI logic has caused the SMI.<br>This will occur when the CPU or a bus master accesses an assigned register (or a sequence of accesses). |
| 20 | 0x0<br>RO/V | **PCI_EXP_SMI Status (PCI_EXP_SMI_STS):** 1- PCI Express SMI event occurred.<br>This could be due to a PCI Express PME event or Hot Plug Event. |
| 19 | 0x0<br>RO/V | **RTC Update-In-Progress SMI Status (RTC_UIP_SMI_STS):** This bit will be set when the RTC Update-In-Progress signal transitions either from low-to-high or high-to-low, depending on the enables in the I/O Trap register space. |
| 18 | 0x0<br>RO/V | **IEH Status (IEH_STS) (IEH_SMI_STS):** This non-sticky read-only bit is a logical OR of each of the SMI status bits in the Internal Error Handler (IEH) Support register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. |
| 17 | 0x0<br>RO/V | **Legacy USB 2 Status (LEGACY_USB2_STS):** This non-sticky read-only bit is a logical OR of each of the SMI status bits in the USB 2.0 Legacy Support Register ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. Writes to this bit will have no effect. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C/V | **SMBUS_SMI Status (SMBUS_SMI_STS):** The SoC sets this bit to '1' to indicate that the SMI# was caused by:<br>• 1. The SMBus Slave receiving a message that an SMI# should be caused, or<br>• 2. The SMBALERT# signal goes active and the SMB_SMI_EN bit is set and the SMBALERT_DIS bit is cleared.<br>• 3. The SMBus Slave receiving a HOST_NOTIFY message and the HOST_NOTIFY_INTREN and the SMB_SMI_EN bits are set.<br>• 4. The SMBus Slave receiving a "SMI in S0" message.<br>This bit is sticky. It is cleared by writing a '1' to this bit position. |
| 15 | 0x0<br>RO/V | **SERIRQ_SMI Status (SERIRQ_SMI_STS):**<br>• 0 = SMI# not caused by SERIRQ decoder<br>• 1 = Indicates the SMI# was caused by the SERIRQ decoder<br>This bit is not sticky. Writes to this bit will have no effect. |
| 14 | 0x0<br>RW/1C/V | **Periodic Status (PERIODIC_STS):** This bit will be set at the rate determined by the PER_SMI_SEL bits.<br>If the PERIODIC_EN bit is also set, the SoC will generate an SMI#.<br>This bit is cleared by writing a '1' to this bit position. |
| 13 | 0x0<br>RW/1C/V | **TCO Status (TCO_STS):**<br>• 0 = SMI not caused by TCO logic.<br>• 1 = Indicates SMI was caused by the TCO logic<br>***Note:*** Will not cause wake event.<br>This bit is cleared by writing a '1' to this bit position. |
| 12 | 0x0<br>RO/V | **DEVMON Status (DEVMON_STS):** This read-only bit is set when bit 0 in the DEVTRAP_STS register is set. It is not sticky, so writes to this bit will have no effect. |
| 11 | 0x0<br>RW/1C/V | **MCSMI Status (MCSMI_STS):** This bit is set if there is an access to the power management microcontroller range (62h or 66h).<br>If this bit is set, and the MCSMI_EN bit is also set, the SoC will generate an SMI#.<br>This bit is set by hardware and cleared by software writing a '1' to its bit position. |
| 10 | 0x0<br>RO/V | **GPIO SMI Status (GPIO_SMI_STS):** This bit will be a 1 if any GPIO that is enabled to trigger SMI is asserted.<br>GPIOs that are not routed to cause an SMI will have no effect on this bit.<br>This bit is NOT sticky. Writes to this bit will have no effect. |
| 9 | 0x0<br>RO/V | **GPE0 Status (GPE0_STS):** There are several status/enable bit pairs in GPE0_STS/EN_127_96 that are capable of triggering SMIs.<br>This bit is a logical OR of all of those pairs (i.e. this bit is asserted whenever at least one of those pairs has both the status and enable bit asserted). This bit is NOT sticky. Writes to this bit will have no effect.<br>***Note:*** The setting of this bit does not cause the SMI#. |
| 8 | 0x0<br>RO/V | **PM1 Status Register (PM1_STS_REG):** This is an OR of the bits (except for bits 5 and 4) in the ACPI PM1 Status Reg. (offset PMBASE+00h).<br>Not sticky. Writes to this bit have no effect.<br>***Note:*** The setting of this bit does not cause the SMI#. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 6 | 0x0<br>RW/1C/V | **Software SMI Timer Status (SWSMI_TMR_STS):** This bit will be set to '1' by the hardware when the Software SMI# Timer expires. This bit will remain 1 until the software writes a '1' to this bit. |
| 5 | 0x0<br>RW/1C/V | **APM Status (APM_STS):** SMI# was generated by a write access to the APM control register and if the APMC_EN bit is set. This bit is cleared by writing a one to its bit position. |
| 4 | 0x0<br>RW/1C/V | **SMI_ON_SLP_EN Status (SMI_ON_SLP_EN_STS):** This bit will be set by the SoC when a write access attempts to set the SLP_EN bit. This bit is cleared by writing a '1' to this bit position |
| 3 | 0x0<br>RO/V | **Legacy USB Status (LEGACY_USB_STS):** This non-sticky read-only bit is a logical OR of each of the SMI status bits in the USB Legacy Keybd Register ANDed with the corresponding enable bits.<br>This bit will not be active if the enable bits are not set. Writes to this bit will have no effect. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/1C/V | **BIOS Status (BIOS_STS):** This bit gets set by hardware when a 1 is written by software to the GBL_RLS bit. When both BIOS_EN and the BIOS_STS bit are set, an SMI# will be generated.<br>The BIOS_STS bit is cleared when software writes a '1' to this bit position. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD_1_0):** Reserved. |

## 58.4.6 General Purpose Event Control (GPE_CTRL)—Offset 40h

Usage: ACPI or Legacy. Not lockable. Primary Power Well.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 40h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_31_18):** Reserved. |
| 17 | 0x0<br>RW/V | **Software GPE Control (SWGPE_CTRL):** This bit allows software to control the assertion of SWGPE_STS bit.<br>This bit is used by hardware as the level input signal for the SWGPE_STS bit in the GPE0_STS register.<br>When SWGPE_CTRL is 1, SWGPE_STS will be set to 1, and writes to SWGPE_STS with a value of 1 to clear SWGPE_STS will result in SWGPE_STS being set back to 1 by hardware.<br>When SWGPE_CTRL is 0, writes to SWGPE_STS with a value of '1' will clear SWGPE_STS to 0. |
| 16:0 | 0x0<br>RO | **Reserved (RSVD_16_0):** Reserved. |

## 58.4.7 Device Trap Status Register (DEVTRAP_STS)—Offset 44h

Usage: Legacy only. Not lockable. Primary Power Well.
Each bit indicates if an access has occurred to the corresponding device's trap range, or for bits 6:9 if the corresponding PCI interrupt is active. Write '1' to the same bit position to clear it. This register is used by APM power management software to see if there has been system activity. The periodic SMI# timer indicates if it is the right time to read the DEVTRAP_STS register.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 44h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:13 | 0x0<br>RO | **Reserved (RSVD_31_13):** Reserved. |
| 12 | 0x0<br>RW/1C/V | **D12 Trap Status (D12_TRP_STS):** KBC (60/64h).<br>• 0 = Indicates that there has been no access to this device I/O range.<br>• 1 = This device I/O range has been accessed.<br>Clear this bit by writing a 1 to the bit location. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_11_10):** Reserved. |
| 9 | 0x0<br>RW/1C/V | **D9 Trap Status (D9_TRP_STS):** PIRQ[D or H].<br>• 0 = The corresponding PCI interrupts have not been active.<br>• 1 = At least one of the corresponding PCI interrupts have been active.<br>Clear this bit by writing a 1 to the bit location. |
| 8 | 0x0<br>RW/1C/V | **D8 Trap Status (D8_TRP_STS):** PIRQ[C or G].<br>• 0 = The corresponding PCI interrupts have not been active.<br>• 1 = At least one of the corresponding PCI interrupts have been active.<br>Clear this bit by writing a 1 to the bit location. |
| 7 | 0x0<br>RW/1C/V | **D7 Trap Status (D7_TRP_STS):** PIRQ[B or F].<br>• 0 = The corresponding PCI interrupts have not been active.<br>• 1 = At least one of the corresponding PCI interrupts have been active.<br>Clear this bit by writing a 1 to the bit location. |
| 6 | 0x0<br>RW/1C/V | **D6 Trap Status (D6_TRP_STS):** PIRQ[A or E].<br>• 0 = The corresponding PCI interrupts have not been active.<br>• 1 = At least one of the corresponding PCI interrupts have been active.<br>Clear this bit by writing a 1 to the bit location. |
| 5 | 0x0<br>RW/1C/V | **D5 Trap Status (D5_TRP_STS):** This will be set if any of the following are accessed (as determined by the I/O ranges in the LPC decoder (even if the LPC forwarding is not enabled): SP1, SP2, PP, FDC.<br>Clear this bit by writing a 1 to the bit location. |
| 4:1 | 0x0<br>RO | **Reserved (RSVD_4_1):** Reserved. |
| 0 | 0x0<br>RW/1C/V | **D0 Trap Status (D0_TRP_STS):** The access/SMI was due to the SATA logic.<br>There is no corresponding bit in the DEVTRAP_EN register because the enables are in the new SATA registers. |

## 58.4.8 PM2a Control Block (PM2A_CNT_BLK)—Offset 50h

Usage: ACPI or Legacy. Not lockable. Primary Power Well.
BIOS must describe this register as 1 byte wide to the OS.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 50h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD_31_1):** Reserved. |
| 0 | 0x0<br>RW | **ARB_DIS:** This bit is a scratchpad bit for legacy software compatibility. |

## 58.4.9 Over-Clocking WDT Control (OC_WDT_CTL)—Offset 54h

This register controls the operation of the SoCs Over-Clocking Watchdog Timer.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 54h

**Default:** 00002000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>WO | **Over-Clocking WDT Reload (OC_WDT_RLD):** Software can write a '1' to this bit to reload (“ping”) the SoC over-clocking watchdog timer while it is running.<br>A write of '0' to this bit has no effect.<br>A write of '1' to this bit while OC_WDT_EN=0, or with the clearing of OC_WDT_EN, does not start or reload the WDT (i.e. OC_WDT_EN takes precedence).<br>The value in OC_WDT_TOV may be changed by software along with its setting of this bit. On a WDT reload, the current (potentially new) value in OC_WDT_TOV is loaded into the WDT as the new timeout value. |
| 30:26 | 0x0<br>RO | **Reserved (RSVD_30_26):** Reserved. |
| 25 | 0x0<br>RW/1C/V | **Over-Clocking WDT ICC Survivability Mode Timeout Status (OC_WDT_ICCSURV_STS):** This bit is set to '1' if the over-clocking WDT has timed out and triggered a global reset while running in a mode that has ICC survivability impact (OC_WDT_ICCSURV=1).<br>It is cleared by a software write of '1' or by the pri_pwrgood_rst_b. |
| 24 | 0x0<br>RW/1C/V | **Over-Clocking WDT Non-ICC Survivability Mode Timeout Status (OC_WDT_NO_ICCSURV_STS):** This bit is set to '1' if the over-clocking WDT has timed out and triggered a global reset while running in a mode that does not have ICC survivability impact (OC_WDT_ICCSURV=0). It is cleared by a software write of '1' or by the pri_pwrgood_rst_b. |
| 23:16 | 0x0<br>RW | **Over-Clocking WDT Scratchpad (OC_WDT_SCRATCH):** This field is available as scratchpad space for software and has no effect on SoC hardware operation. |
| 15 | 0x0<br>RW/L | **Over-Clocking WDT Force All (OC_WDT_FORCE_ALL):**<br>GATE_BIT: OC_WDT_CTL.OC_WDT_CTL_LCK.HIGH<br>When this bit is set to 1 and the OC_WDT is running, any included global reset source will behave as though the OC_WDT expired. |
| 14 | 0x0<br>RW/V/L | **Over-Clocking WDT Enable (OC_WDT_EN):** Software sets this bit to '1' to enable the SoC over-clocking watchdog timer. While the counter is running, if it expires before being reloaded by software via the OC_WDT_RLD bit or halted by software clearing this bit, then one of the status bits will be set (which one depends on the WDT operating mode at the time - see the OC_WDT_ICCSURV bit description), and a global reset will be triggered.<br>This bit is also set by the hardware when conditions allow the OC_WDT to self-start at power cycle reboot (effectively changes the default of this bit as seen by software). |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 13 | 0x1<br>RW/L | **Over-Clocking WDT ICC Survivability Impact (OC_WDT_ICCSURV):** This bit determines whether OC_WDT expiration will have an impact on ICC (Integrated Clock Controller) bootstrap survivability.<br>• 0 = Software should configure the OC_WDT to this mode if no ICC hardware auto-recovery actions are desired in the event of a timeout. A timeout in this mode will set OC_WDT_NO_ICCSURV_STS.<br>• 1 (Default) = An OC_WDT timeout while operating in this mode causes certain ICC hardware auto-recovery actions to take place. A timeout in this mode will set OC_WDT_ICCSURV_STS. |
| 12 | 0x0<br>RW/L | **OC_WDT_CTL Register Lock (OC_WDT_CTL_LCK):** This bit controls write-ability to this register. Encodings:<br>• 0 = All fields of register OC_WDT_CTL operate as normal and can be updated by software. Reads to the register operate as normal.<br>• 1 = All RW/L fields of register OC_WDT_CTL, including this lock control bit, are locked. Writes to these register fields have no effect and the register fields retain their current states. Reads to the register operate as normal.<br>Once this bit is set, it can only be cleared by CORE well power loss. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_11_10):** Reserved. |
| 9:0 | 0x0<br>RW/V/L | **Over-Clocking WDT Timeout Value (OC_WDT_TOV):** Software programs the desired over-clocking WDT timeout value into this register. This timer is zero-based and has a granularity of 1 second. Example timeout values:<br>• 000h: 1 second<br>• 001h: 2 seconds<br>• ...<br>• 3FFh: ~17 minutes (1024 seconds)<br>The value of OC_WDT_TOV may be changed by software along with its setting of the OC_WDT_RLD bit. On a WDT reload, the current (potentially new) value in OC_WDT_TOV is loaded into the WDT as the new timeout value.<br>This field is also updated by the hardware when conditions allow the OC_WDT to self-start at power cycle reboot (effectively changes the default of this field as seen by software). |

## 58.4.10 General Purpose Event 0 Status [31:0] (GPE0_STS_31_0)—Offset 80h

GPE0_STS bits 31:0 are assigned to certain SoC GPIO signals as GPE0_DW0. This register is symmetrical to the General Purpose Event 0 Enable [31:0] Register.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 80h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/1C/V | **General Purpose Event 0 Status [31:0] (GPE0_STS_31_0):** These bits are mapped to the SoC General-Purpose I/O (GPIO) signals that can generate GPE0 when properly configured. |

## 58.4.11 General Purpose Event 0 Status [63:32] (GPE0_STS_63_32)—Offset 84h

GPE0_STS bits 63:32 are assigned to certain SoC GPIO signals as GPE0_DW1. This register is symmetrical to the General Purpose Event 0 Enable [63:32] Register.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 84h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/1C/V | **General Purpose Event 0 Status [63:32] (GPE0_STS_63_32):** These bits are mapped to the SoC General-Purpose I/O (GPIO) signals that can generate GPE0 when properly configured. |

## 58.4.12 General Purpose Event 0 Status [95:64] (GPE0_STS_95_64)—Offset 88h

GPE0_STS bits 95:64 are assigned to certain SoC GPIO signals as GPE0_DW2. This register is symmetrical to the General Purpose Event 0 Enable [95:64] Register.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 88h

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/1C/V | **General Purpose Event 0 Status [95:64] (GPE0_STS_95_64):** These bits are mapped to the SoC General-Purpose I/O (GPIO) signals that can generate GPE0 when properly configured. |

## 58.4.13 General Purpose Event 0 Status [127:96] (GPE0_STS_127_96)—Offset 8Ch

This register is symmetrical to the General Purpose Event 0 Enable [127:96] Register. Unless indicated otherwise below, if the corresponding _EN bit is set, then when the STS bit get set, the SoC generates a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the SoC also generates an SCI if the SCIEN bit is set, or an SMI# if the SCIEN bit is not set and GBL_SMI_EN is set.

**Type:** I/O Register
(Size: 32 bits)

**BAR and Offset:** [ABASE] + 8Ch

**Default:** 00000000h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Reserved (RSVD_31_19):** Reserved. |
| 18 | 0x0<br>RW/1C/V | **Wake Alarm Device Timer Status (WADT_STS):** This bit is set whenever the any of the wake alarm device timers signal a timer expiration. |
| 17 | 0x0<br>RW/1C/V | **USB Connection in/after DeepSx Status (USB_CON_DSX_STS):** This bit is set when a connection event occurs on any unmasked USB port while in DeepSx or while in Sx immediately after DeepSx.<br>***Note:*** This bit will be set regardless of the value of the DSX_CFG.USB_CON_DSX_MODE bit. |
| 16 | 0x0<br>RW/1C/V | **GPIO[27] Status (GP27_STS):** This bit is set whenever GPIO[27] is seen asserted low.<br>Note that unlike the GPI[n]_STS bits, GPIO[27] is always monitored as an input for the purpose of setting this bit, regardless of the actual GPIO configuration. |
| 15 | 0x0<br>RW/1C/V | **GPIO Tier2 SCI Status (GPIO_TIER2_SCI_STS):** This bit is a logical OR of sci_wake from tier 2 GPIO's.<br>reset_type = host_deep_rst_b |
| 14 | 0x0<br>RW/1C/V | **Reserved:** Reserved. |
| 13 | 0x0<br>RW/1C/V | **Power Management Event Bus 0 Status (PME_B0_STS):** This bit will be set to 1 by the SoC when any internal device with PCI Power Management capabilities on bus 0 asserts the equivalent of the PME# signal. Additionally, if the PME_B0_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the PME_B0_STS bit will generate an SCI (or SMI# if SCI_EN is not set).<br>If the PME_B0_EN bit is set, and the system is in an S1-S4 state (or S5 state due to SLP_TYP and SLP_EN), then the setting of the PME_B0_STS bit will generate a wake event. If the system is in an S5 state due to power button override, then the PME_B0_STS bit will not cause a wake event or SCI.<br>This bit is cleared by a software write of '1'.<br>Internal devices which can set this bit:<br>• Integrated LAN<br>• SATA<br>• EHCI (USB 2.0)<br>• XHCI (USB 3.0)<br>• Intel® ME Maskable Host Wake |
| 12 | 0x0<br>RW/1C/V | **ME SCI Status (ME_SCI_STS):** This bit will be set when the Intel® ME is requesting an SCI. Software must clear the ME source of the SCI before clearing this bit.<br>***Note:*** This source is not able to cause a wake event. |
| 11 | 0x0<br>RW/1C/V | **Power Management Event Status (PME_STS):** This bit will be set to 1 by hardware when the PME# signal goes active.<br>Additionally, if the PME_EN and SCI_EN bits are set, and system is in an S0 state, then the setting of the PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set).<br>If the PME_EN bit is set, and the system is in an S1-S4 state (or S5 state due to SLP_TYP and SLP_EN), then the setting of the PME_STS bit will generate a wake event. If the system is in an S5 state due to power button override, then PME_STS will not cause a wake event or SCI.<br>This bit is cleared by writing a 1 to this bit position. |
| 10 | 0x0<br>RO | **Reserved (RSVD_10):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 9 | 0x0<br>RW/1C/V | **PCI Express Status (PCI_EXP_STS):** This bit will be set to 1 by hardware to indicate that:<br>• The PME event message was received on one or more of the PCI-Express Ports<br>• An Assert PMEGPE message received from the MCH via DMI<br>***Note:*** The PCI WAKE# pin and the PCI-Express Beacons have no impact on this bit.<br>Software attempts to clear this bit by writing a 1 to this bit position. If the PCI_EXP_STS bit went active due to an Assert PMEGPE message, then a Deassert PMEGPE message must be received prior to the software write in order for the bit to be cleared.<br>If the bit is not cleared and the corresponding PCI_EXP_EN bit is set, the level-triggered SCI will remain active. |
| 8 | 0x0<br>RW/1C/V | **RI Status (RI_STS):** This bit will be set to1 by hardware when the RI# input signal goes active. This bit can be reset by writing a 1 to this bit position. |
| 7 | 0x0<br>RW/1C/V | **SMBus Wake Status (SMB_WAK_STS):** This bit is set to 1 by the hardware to indicate that the wake event was caused by the SoC's SMBus logic. This could be due to either the SMBus slave unit receiving a message or the SMBALERT# signal going active. |
| 6 | 0x0<br>RW/1C/V | **TCOSCI Status (TCOSCI_STS):** This bit will be set to 1 by hardware when the TCO logic or Thermal Sensor logic causes an SCI. This bit can be reset by writing a one to this bit position. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD_5_4):** Reserved. |
| 3 | 0x0<br>RW/1C/V | **IE SCI Status (IE_SCI_STS):** This bit will be set when the Innovative Engine (IE) is requesting an SCI. Software must clear the IE source of the SCI before clearing this bit.<br>***Note:*** This source is not able to cause a wake event. |
| 2 | 0x0<br>RW/1C/V | **Software GPE Status (SWGPE_STS):** The SWGPE_CTRL bit (bit 1 of GPE_CTRL reg) acts as a level input to this bit. |
| 1 | 0x0<br>RW/1C/V | **Hot Plug Status (HOT_PLUG_STS):** This bit is set to 1 by hardware when a PCI-Express hot-plug event occurs. This will cause an SCI if the HOT_PLUG_EN and SCI_EN bits are set. This bit is cleared by writing a '1' to this bit position. The following events cause this bit to set:<br>• Assert_GPE message received from any of the PCI Express ports in the SoC<br>• Assert_HPGPE message received from any of the PCI Express ports in the SoC |
| 0 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |

## 58.4.14 General Purpose Event 0 Enable [31:0] (GPE0_EN_31_0)—Offset 90h

GPE0_EN bits 31:0 are assigned to certain SoC GPIO signals as GPE0_DW0. This register is symmetrical to the General Purpose Event 0 Status [31:0] Register.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 90h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **General Purpose Event 0 Enable [31:0] (GPE0_EN_31_0):** See Chapter 26, "Customer General-Purpose I/O (GPIO)." |

## 58.4.15 General Purpose Event 0 Enable [63:32] (GPE0_EN_63_32)—Offset 94h

GPE0_EN bits 63:32 are assigned to certain SoC GPIO signals as GPE0_DW1. This register is symmetrical to the General Purpose Event 0 Status [63:32] Register.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 94h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **General Purpose Event 0 Enable [63:32] (GPE0_EN_63_32):** See Chapter 26, "Customer General-Purpose I/O (GPIO)." |

## 58.4.16 General Purpose Event 0 Enable [95:64] (GPE0_EN_95_64)—Offset 98h

GPE0_EN bits 95:64 are assigned to certain SoC GPIO signals as GPE0_DW2. This register is symmetrical to the General Purpose Event 0 Status [95:64] Register.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 98h

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **General Purpose Event 0 Enable [95:64] (GPE0_EN_95_64):** See Chapter 26, "Customer General-Purpose I/O (GPIO)." |

## 58.4.17 General Purpose Event 0 Enable [127:96] (GPE0_EN_127_96)—Offset 9Ch

This register is symmetrical to the General Purpose Event 0 Status [127:96] Register.

**Type:** I/O Register
(Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [ABASE] + 9Ch

**ABASE Type:** PCI Configuration Register (Size: 32 bits)
**ABASE Reference:** [B:0, D:31, F:2] + 40h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:19 | 0x0<br>RO | **Reserved (RSVD_31_19):** Reserved. |
| 18 | 0x0<br>RW | **Wake Alarm Device Timer Enable (WADT_EN):** Used to enable the setting of the WADT_STS bit to generate wake/SMI#/SCI. |
| 17 | 0x0<br>RW | **USB Connection in/after DeepSx Enable (USB_CON_DSX_EN):** Used to enable the setting of the USB_CON_DSX_STS bit to generate wake/SMI#/SCI. |
| 16 | 0x0<br>RW | **GPIO[27] Enable (GP27_EN):** Used to enable the setting of the GP27_STS bit to generate wake/SMI#/SCI. Host wake events from the PHY through GP27 cannot be disabled by clearing this bit.<br>***Note:*** GPIO[27] is a valid host wake event from Deep-Sx. But the wake enable configuration must persist even after a G3. To make this happen, this bit is in the SoC RTC power well. |
| 15 | 0x0<br>RW/V | **GPIO Tier2 SCI EN (GPIO_TIER2_SCI_EN):** Used to enable the setting of GPIO_TIER2_SCI_STS to generate wake/SCI#<br>reset_type =host_deep_rst_b |
| 14 | 0x0<br>RW/V | **Reserved:** Reserved. |
| 13 | 0x0<br>RW/V | **PME_B0 Enable (PME_B0_EN):** Enables the setting of the PME_B0_STS bit to generate a wake event and/or an SCI or SMI#. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 12 | 0x0<br>RW/V | **ME SCI Enable (ME_SCI_EN):** Used to enable the setting of the ME_SCI_STS bit to generate a SCI. |
| 11 | 0x0<br>RW/V | **Power Management Event Enable (PME_EN):** Enables the setting of the PME_STS to generate a wake event and/or an SCI. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 10 | 0x0<br>RO | **Reserved (RSVD_10):** Reserved. |
| 9 | 0x0<br>RW/V | **PCI Express Enable (PCI_EXP_EN):** Enables the SoC to cause an SCI when PCI_EXP_STS bit is set. This is used to allow the PCI Express ports to cause an SCI due to wake/PME events. |
| 8 | 0x0<br>RW/V | **RI Enable (RI_EN):** When RI_EN and RI_STS are both set, a Wake event will occur.<br>If RI_EN is not set, then when RI_STS is set, no Wake event will occur. This bit is in the RTC power well and is cleared when the SoC input signal pin SRTCRST_N is active. |
| 7 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved.<br>***Note:*** For the SoC, the SMBus slave will always be enabled as a wake event, as will the SMBALERT# signal. |
| 6 | 0x0<br>RW/V | **TCOSCI Enable (TCOSCI_EN):** When TCOSCI_EN and TCOSCI_STS are both set, an SCI will be generated. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD_5_4):** Reserved. |
| 3 | 0x0<br>RW/V | **IE SCI Enable (IE_SCI_EN):** Used to enable the setting of the IE_SCI_STS bit to generate a SCI. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW/V | **Software GPE Enable (SWGPE_EN):** This bit, when set to 1, enables the SW GPE function.<br>• If SWGPE_CTRL is written to a 1, hardware will set SWGPE_STS (acts as a level input).<br>• If SWGPE_STS, SWGPE_EN, and SCI_EN are all 1's, an SCI will be generated.<br>• If SWGPE_STS = 1, SWGPE_EN = 1, SCI_EN = 0, and GBL_SMI_EN = 1, then an SMI# will be generated. |
| 1 | 0x0<br>RW/V | **Hot Plug Enable (HOT_PLUG_EN):** Enables the SoC to cause an SCI when the HOT_PLUG_STS bit is set. This is used to allow the PCI Express ports to cause an SCI due to hot-plug events. The following events cause HOT_PLUG_STS bit to set:<br>• Assert_GPE message received from any of the PCI Express ports in the SoC<br>• Assert_HPGPE message received from any of the PCI Express ports in the SoC |
| 0 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |

## 58.5 Registers in I/O Space—Fixed Addresses

This register is accessed in Host I/O Space through a fixed I/O address.

### 58.5.1 Reset Control Register (RST_CNT)

**Type:** I/O Register
(Size: 8 bits)

**Fixed Address in Host I/O Space:** CF9h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:4 | 0x0<br>RO | **Reserved (RSVD_7_4):** Reserved. |
| 3 | 0x0<br>RW | **Full Reset (FULL_RST):** When this bit is set to 1 and bit 1 is set to 1 (indicating Hard Reset, not Soft Reset), and the RST_CPU bit (bit 2) is written from 0 to 1, the SoC performs a full reset, including driving signal pins PMU_SLP_S3_N and PMU_SLP_S45_N active (low) for at least 3, and no more than 5, seconds.<br>When this bit is set, it also causes the full power cycle (PMU_SLP_S3_N and PMU_SLP_S45_N assertion) in response to PMU_RESETBUTTON_N, COREPWROK, and Watchdog Timer reset sources. |
| 2 | 0x0<br>RW | **Reset CPU (RST_CPU):** This bit will cause either a hard or soft reset to the CPU depending on the state of the SYS_RST bit (bit 1 in this same register).<br>The software will cause the reset by setting bit 2 from a 0 to a 1. |
| 1 | 0x0<br>RW | **System Reset (SYS_RST):** This bit determines the type of reset caused via RST_CPU (bit 2 of this register).<br>• If SYS_RST is 0 when RST_CPU goes from 0 to 1, then the SoC will force an internal Initialize (INIT#) signal to the CPU but system can hang due to the security concern with this internal initialize (INIT#) enabled.<br>• If SYS_RST is 1 when RST_CPU goes from 0 to 1, then the SoC will force PCI Reset active for about 1 ms, however the PMU_SLP_S3_N and PMU_SLP_S45_N signals assertions are dependent on the value of the FULL_RST (bit 3 of this register). |
| 0 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |

## 58.6 Sideband Registers

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 58.6.1 Advanced Power Management Status and Control Port (APM_STS_CNT)—Offset B0h

Usage: Legacy only. Not lockable. Primary Power Well.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x82] + B0h
**Host Memory Space:** SBREG_BAR + 0x820000 + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW | **APM Status Port (APM_STS):** Used to pass data between the OS and the SMI handler.<br>Basically, this is a scratchpad register and is not effected by any other register or function (other than a PCI reset). |
| 23:16 | 0x0<br>RW | **Advanced Power Management Control Port (APM_CNT):** Used to pass an APM command between the OS and the SMI handler.<br>Writes to this port not only store data in the APMC register, but also generates an SMI# when the APMC_EN bit is set. |
| 15:0 | 0x0<br>RO | **Reserved (RSVD_15_0):** Reserved. |

### 58.6.2 Misc SB Reg0 (MISC_SBR0)—Offset B4h

Misc SB Reg0

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0x82] + B4h
**Host Memory Space:** SBREG_BAR + 0x820000 + B4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_31_9):** Reserved. |
| 8 | 0x0<br>RW | **North Completed (NORTH_COMPLETED):** North Completed<br>• 0: Punit has not completed north preparation for Asynchronous DRAM Refresh (ADR)<br>• 1: Punit has completed north preparation for ADR<br>Set by Pcode through CRWt once Pcode has put DIMM in Self-refresh. |
| 7:5 | 0x0<br>RO | **Reserved (RSVD_7_5):** Reserved. |
| 4:1 | 0x0<br>RO/V | **FINAL DDR SPEED set by PMC FW (FINAL_DDR_SPEED):** FINAL_DDR_SPEED set by PMC firmware (CURRENT_SPD_SPEED):<br>• 0xxxx = undefined, use max DDR mem config (fuse) as IDI speed<br>• 1000 = DDR1600, 400MHz IDI freq<br>• 1001 = DDR1866, 466MHz IDI freq<br>• 1010 = DDR2133, 533MHz IDI freq<br>• 1011 = DDR2400, 600Mhz IDI freq<br>• 1100 = DDR2666, 667MHz IDI freq |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RW/1C/V | **SPD_RESET_PCODE:** SPD_RESET_PCODE (SPD_RESET_PCODE)<br>• 0= PMC indicates to Pcode that this is NOT a SPD Reset.<br>• 1= PMC indicates to Pcode that this is a SPD ResetSet by PMC firmware during the SPD Reset clocking flow. |

§ §

# 59 SMBus - Legacy - B0, D31, F4

## 59.1 Introduction and Index

The host-accessible registers for the SMBus are described here.

To determine the number of devices/functions available refer to the following:

- Table 1-1, “Features by product SKU - Server and Cloud Storage SKUs 0 through 4”
- Table 1-2, “Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10”
- Table 1-3, “Features by Product SKU - Extended Temperature SKUs 11 through 14”.

The legend for the register access codes are described in Chapter 43, “Introduction.”

### 59.1.1 Registers in Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 4. The offset addresses are listed.

**Table 59-1. Summary of PCI Configuration Registers—0/31/4**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 8086 | "Vendor ID (VID)—Offset 0h" on page 2755 |
| 2 | 2 | 19DF | "Device ID (DID)—Offset 2h" on page 2755 |
| 4 | 2 | 0000 | "Command (CMD)—Offset 4h" on page 2756 |
| 6 | 2 | 0280 | "Device Status (DS)—Offset 6h" on page 2757 |
| 8 | 1 | 00 | "Revision ID (RID)—Offset 8h" on page 2758 |
| 9 | 1 | 00 | "Programming Interface (PI)—Offset 9h" on page 2758 |
| A | 1 | 05 | "Sub Class Code (SCC)—Offset Ah" on page 2758 |
| B | 1 | 0C | "Base Class Code (BCC)—Offset Bh" on page 2758 |
| 10 | 4 | 00000004 | "SMBus Memory Base Address_31_0 (SMBMBAR_31_0)—Offset 10h" on page 2759 |
| 14 | 4 | 00000000 | "SMBus Memory Base Address_63_32 (SMBMBAR_63_32)—Offset 14h" on page 2759 |
| 20 | 4 | 00000001 | "SMB Base Address (SBA)—Offset 20h" on page 2760 |
| 2C | 2 | 0000 | "SVID—Offset 2Ch" on page 2760 |
| 2E | 2 | 0000 | "SID—Offset 2Eh" on page 2760 |
| 3C | 1 | 00 | "Interrupt Line (INTLN)—Offset 3Ch" on page 2761 |
| 3D | 3 | 01 | "Interrupt Pin (INTPN)—Offset 3Dh" on page 2761 |
| 40 | 1 | 00 | "Host Configuration (HCFG)—Offset 40h" on page 2762 |
| 50 | 4 | 00000001 | "TCO Base Address (TCOBASE)—Offset 50h" on page 2763 |
| 54 | 4 | 00000000 | "TCO Control (TCOCTL)—Offset 54h" on page 2763 |
| 64 | 4 | 08080000 | "Host_Timing (HTIM)—Offset 64h" on page 2764 |
| F8 | 4 | 01000F1C | "Manufacturer's ID (MANID)—Offset F8h" on page 2765 |

### 59.1.2 Registers in Memory Space—SMBMBAR

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 59-2. Summary of Memory Mapped I/O Registers—SMBMBAR**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 00 | "Host Status Register Address (HSTS)—Offset 0h" on page 2766 |
| 2 | 1 | 00 | "Host Control Register (HCTL)—Offset 2h" on page 2767 |
| 3 | 1 | 00 | "Host Command Register (HCMD)—Offset 3h" on page 2769 |
| 4 | 1 | 00 | "Transmit Slave Address Register (TSA)—Offset 4h" on page 2769 |
| 5 | 1 | 00 | "Data 0 Register (HD0)—Offset 5h" on page 2769 |
| 6 | 1 | 00 | "Data 1 Register (HD1)—Offset 6h" on page 2770 |
| 7 | 1 | 00 | "Host Block Data (HBD)—Offset 7h" on page 2770 |
| 8 | 1 | 00 | "Packet Error Check Data Register (PEC)—Offset 8h" on page 2771 |
| 9 | 1 | 44 | "Receive Slave Address Register (RSA)—Offset 9h" on page 2771 |
| A | 2 | 0000 | "Slave Data Register (SD)—Offset Ah" on page 2772 |
| C | 1 | 00 | "Auxiliary Status (AUXS)—Offset Ch" on page 2773 |
| D | 1 | 00 | "Auxiliary Control (AUXC)—Offset Dh" on page 2774 |
| E | 1 | 04 | "SMLINK_PIN_CTL Register (SMLC)—Offset Eh" on page 2775 |
| F | 1 | 04 | "SMBUS_PIN_CTL Register (SMBC)—Offset Fh" on page 2776 |
| 10 | 1 | 00 | "Slave Status Register (SSTS)—Offset 10h" on page 2777 |
| 11 | 3 | 00 | "Slave Command Register (SCMD)—Offset 11h" on page 2778 |
| 14 | 2 | 00 | "Notify Device Address Register (NDA)—Offset 14h" on page 2779 |
| 16 | 1 | 00 | "Notify Data Low Byte Register (NDLB)—Offset 16h" on page 2779 |
| 17 | 1 | 00 | "Notify Data High Byte Register (NDHB)—Offset 17h" on page 2779 |

### 59.1.3 Registers in I/O Space—SBA

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 59-3. Summary of I/O Registers—SBA**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 00 | "Host Status Register Address (HSTS)—Offset 0h" on page 2780 |
| 2 | 1 | 00 | "Host Control Register (HCTL)—Offset 2h" on page 2781 |
| 3 | 1 | 00 | "Host Command Register (HCMD)—Offset 3h" on page 2783 |
| 4 | 1 | 00 | "Transmit Slave Address Register (TSA)—Offset 4h" on page 2783 |
| 5 | 1 | 00 | "Data 0 Register (HD0)—Offset 5h" on page 2783 |
| 6 | 1 | 00 | "Data 1 Register (HD1)—Offset 6h" on page 2784 |
| 7 | 1 | 00 | "Host Block Data (HBD)—Offset 7h" on page 2784 |
| 8 | 1 | 00 | "Packet Error Check Data Register (PEC)—Offset 8h" on page 2785 |
| 9 | 1 | 44 | "Receive Slave Address Register (RSA)—Offset 9h" on page 2785 |
| A | 2 | 0000 | "Slave Data Register (SD)—Offset Ah" on page 2786 |
| C | 1 | 00 | "Auxiliary Status (AUXS)—Offset Ch" on page 2787 |
| D | 1 | 00 | "Auxiliary Control (AUXC)—Offset Dh" on page 2788 |
| E | 1 | 04 | "SMLINK_PIN_CTL Register (SMLC)—Offset Eh" on page 2789 |
| F | 1 | 04 | "SMBUS_PIN_CTL Register (SMBC)—Offset Fh" on page 2790 |
| 10 | 1 | 00 | "Slave Status Register (SSTS)—Offset 10h" on page 2791 |
| 11 | 3 | 00 | "Slave Command Register (SCMD)—Offset 11h" on page 2792 |
| 14 | 2 | 00 | "Notify Device Address Register (NDA)—Offset 14h" on page 2793 |
| 16 | 1 | 00 | "Notify Data Low Byte Register (NDLB)—Offset 16h" on page 2793 |
| 17 | 1 | 00 | "Notify Data High Byte Register (NDHB)—Offset 17h" on page 2793 |

### 59.1.4 Registers in I/O Space—TCOBASE

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 59-4. Summary of I/O Registers—TCOBASE**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 2 | 0004 | "TCO_RLD Register (TRLD)—Offset 0h" on page 2794 |
| 2 | 1 | 00 | "TCO_DAT_IN Register (TDI)—Offset 2h" on page 2794 |
| 3 | 1 | 00 | "TCO_DAT_OUT Register (TDO)—Offset 3h" on page 2795 |
| 4 | 2 | 0000 | "TCO1_STS Register (TSTS1)—Offset 4h" on page 2795 |
| 6 | 2 | 0000 | "TCO2_STS Register (TSTS2)—Offset 6h" on page 2797 |
| 8 | 2 | 0000 | "TCO1_CNT Register (TCTL1)—Offset 8h" on page 2798 |
| A | 2 | 0008 | "TCO2_CNT Register (TCTL2)—Offset Ah" on page 2799 |
| 10 | 1 | 03 | "LEGACY_ELIM Register (LE)—Offset 10h" on page 2800 |
| 12 | 2 | 0004 | "TCO_TMR Register (TTMR)—Offset 12h" on page 2800 |

### 59.1.5 Sideband Registers—Host SMBus Private Configuration

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 59-5. Summary of Sideband Registers—0xCF**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 0 | 00000000 | "TCO Configuration (TCOCFG)—Offset 0h" on page 2801 |
| C | 00000000 | "General Control (GC)—Offset Ch" on page 2802 |
| 10 | 00000009 | "Power Control Enable (PCE)—Offset 10h" on page 2803 |

### 59.1.6 Sideband Registers—CLTT SMBus Configuration Control

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 59-6. Summary of Sideband Registers—0xCC**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 10 | 00000000 | "Ch0 Dimm0 Cfg Status (C0D0CFGSTS)—Offset 10h" on page 2805 |
| 14 | 01000000 | "Ch0 Dimm1 Cfg Status (C0D1CFGSTS)—Offset 14h" on page 2805 |
| 18 | 00000000 | "Ch0 Dimm2 Cfg Status (C0D2CFGSTS)—Offset 18h" on page 2806 |
| 1C | 00000000 | "Ch0 Dimm3 Cfg Status (C0D3CFGSTS)—Offset 1Ch" on page 2806 |
| 20 | 02000000 | "Ch1 Dimm0 Cfg Status (C1D0CFGSTS)—Offset 20h" on page 2807 |
| 24 | 03000000 | "Ch1 Dimm1 Cfg Status (C1D1CFGSTS)—Offset 24h" on page 2807 |
| 28 | 00000000 | "Ch1 Dimm2 Cfg Status (C1D2CFGSTS)—Offset 28h" on page 2808 |
| 2C | 00000000 | "Ch1 Dimm3 Cfg Status (C1D3CFGSTS)—Offset 2Ch" on page 2808 |
| 30 | 00000000 | "Ch0 Dimm Temp Status (C0TEMPSTS0)—Offset 30h" on page 2809 |
| 34 | 00000000 | "Ch0 Dimm Temp Status (C0TEMPSTS1)—Offset 34h" on page 2809 |
| 38 | 00000000 | "Ch1 Dimm Temp Status (C1TEMPSTS0)—Offset 38h" on page 2809 |
| 3C | 00000000 | "Ch1 Dimm Temp Status (C1TEMPSTS1)—Offset 3Ch" on page 2810 |
| 40 | 00 | "Host Configuration (HCFG)—Offset 40h" on page 2810 |

## 59.2 Registers in Configuration Space

### 59.2.1 Vendor ID (VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 0h

**Default:** 8086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x8086<br>RO | **Vendor ID (VID):** Value indicates Intel as the vendor.<br><br>**Power Well:** Core |

### 59.2.2 Device ID (DID)—Offset 2h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 2h

**Default:** 19DFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x19df<br>RO/V | **Device ID (DID):** Indicates the device number assigned by the SIG.<br><br>**Power Well:** Core |

## 59.2.3 Command (CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 4h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:11 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **Interrupt Disable (INTD):**<br>• 1 = Disables SMBus to assert its PIRQB# signal.<br>Defaults to 0.<br><br>**Power Well:** Core |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBE):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **SERR# Enable (SERRE):**<br>• 1 = Enables SERR# generation.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Wait Cycle Control (WCC):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **Parity Error Response (PER):**<br>• 1 = Sets Detected Parity Error bit when parity error is detected.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Postable Memory Write Enable (PMWE):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Special Cycle Enable (SCE):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **Bus Master Enable (BME):** Reserved as 0.<br>Read Only.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):**<br>• 1= Enables memory mapped config space.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **I/O Space Enable (IOSE):**<br>• 1= enables access to the SM Bus I/O space registers as defined by the Base Address Register.<br><br>**Power Well:** Core |

## 59.2.4 Device Status (DS)—Offset 6h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 6h

**Default:** 0280h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/1C | **Detected Parity Error (DPE):** Reserved.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RW/1C | **Signaled System Error (SSE):** Reserved.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RO | **Received Master Abort (RMA):** Reserved as 0.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO | **Received Target Abort (RTA):** Reserved as 0.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Signaled Target-Abort Status (STA):** Reserved as 0.<br><br>**Power Well:** Core |
| 10:9 | 0x1<br>RO | **DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RO | **Data Parity Error Detected (DPED):** Reserved as 0.<br><br>**Power Well:** Core |
| 7 | 0x1<br>RO | **Fast Back-to-Back Capable (FBC):** Reserved as 1.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RO | **User Definable Features (UDF):** Reserved as 0.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **66 MHz Capable (C_66M):** Reserved as 0.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Capabilities List Indicator (CLI):** Hardwired to 0 because there are no capability list structures in this function.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Interrupt Status (INTS):** This bit indicates that an interrupt is pending.<br>It is independent from the state of the Interrupt Enable bit in the command register.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |

## 59.2.5 Revision ID (RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Revision ID (RID):** The value reported in this register depends on the global revision ID.<br><br>**Power Well:** Core |

## 59.2.6 Programming Interface (PI)—Offset 9h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + 9h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **Programming Interface (PI):** Programming interface.<br><br>**Power Well:** Core |

## 59.2.7 Sub Class Code (SCC)—Offset Ah

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + Ah

**Default:** 05h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x5<br>RO | **Sub Class Code (SCC):** A value of 05h indicates that this device is a SM Bus serial controller.<br><br>**Power Well:** Core |

## 59.2.8 Base Class Code (BCC)—Offset Bh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + Bh

**Default:** 0Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0xc<br>RO | **Base Class Code (BCC):** A value of 0Ch indicates that this device is a serial controller.<br><br>**Power Well:** Core |

## 59.2.9 SMBus Memory Base Address_31_0 (SMBMBAR_31_0)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 10h

**Default:** 00000004h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RW | **Base Address (BA):** Provides the 32 byte system memory base address.<br><br>**Power Well:** Core |
| 7:4 | 0x0<br>RO | **Hardwired_0:** Hardwired to 0.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Prefetchable (PREF):** Hardwired to 0.<br>Indicated that SMBMBAR is not prefetchable.<br><br>**Power Well:** Core |
| 2:1 | 0x2<br>RO | **Address Range (ADDRNG):** Indicates that this SMBMBAR can be located anywhere in 64 bit address space.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Memory Space Indicator (MSI):** Indicates that the SMB logic is memory mapped.<br><br>**Power Well:** Core |

## 59.2.10 SMBus Memory Base Address_63_32 (SMBMBAR_63_32)—Offset 14h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 14h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **Base Address (BA):** Bits 63-32 of SMBus Memory Base Address.<br><br>**Power Well:** Core |

## 59.2.11 SMB Base Address (SBA)—Offset 20h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 20h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:5 | 0x0<br>RW | **Base Address (BA):** Provides the 32 byte t system I/O base address for the SMB logic.<br><br>**Power Well:** Core |
| 4:1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RO | **IO Space Indicator (IOSI):** This read-only bit always is 1, indicating that the SMB logic is I/O mapped.<br><br>**Power Well:** Core |

## 59.2.12 SVID—Offset 2Ch

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 2Ch

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/O | **SVID:** BIOS sets the value in this register to identify the Subsystem Vendor ID.<br>The SMBus SVID register, in combination with the SMBus Subsystem ID register, enables the operating system to distinguish each subsystem from the others.<br>***Note:*** The software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle.<br><br>**Power Well:** Core |

## 59.2.13 SID—Offset 2Eh

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:4] + 2Eh

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RW/O | **SID:** BIOS can write to this register to identify the Subsystem ID.<br>The SID register, in combination with the SVID, enable the operating system to distinguish each subsystem from other(s).<br>***Note:*** The software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle.<br><br>**Power Well:** Core |

## 59.2.14 Interrupt Line (INTLN)—Offset 3Ch

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + 3Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Line (INTLN):** This data is not used by the hardware.<br>It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.<br><br>**Power Well:** Core |

## 59.2.15 Interrupt Pin (INTPN)—Offset 3Dh

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + 3Dh

**Default:** 01h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x1<br>RO | **Interrupt Pin (INTPN):** This defines the interrupt pin to be used by the SMBus controller.<br>Bits: Pins<br>• 0h: No Interrupt<br>• 1h: INTA#<br>• 2h: INTB#<br>• 3h: INTC#<br>• 4h: INTD#<br>• 5h-Fh: Reserved BIOS<br><br>**Power Well:** Core |

## 59.2.16 Host Configuration (HCFG)—Offset 40h

**Type:** PCI Configuration Register (Size: 8 bits)

**Offset:** [B:0, D:31, F:4] + 40h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/1L | **SPD Write Disable (SPDWD):** No SMB registers should be accessed while HSTS.HBSY bit is set, the same will apply to SPD write disable.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **SSRESET:** Soft SMBUS Reset: When this bit is 1, the SMBus state machine and logic is reset. The HW will reset this bit to 0 when reset operation is completed.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **I2C_EN (I2CEN):** When this bit is 1, the SMBus Host is enabled to communicate with I2C devices. This will change the formatting of some commands. When this bit is 0, behavior is for SMBus.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **SMB_SMI_EN (SSEN):** When this bit is set, any source of an SMB interrupt will instead be routed to generate an SMI#.<br>Refer to Section 28.8, "Interrupts and SMI" on page 722 (Interrupts/SMI#) for details.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **HST_EN (HSTEN):** When set, the SMB Host Controller interface is enabled to execute commands. The HST_INT_EN bit needs to be enabled in order for the SMB Host Controller to interrupt or SMI#. Additionally, the SMB Host Controller will not respond to any new requests until all interrupt requests have been cleared.<br><br>**Power Well:** Core |

## 59.2.17 TCO Base Address (TCOBASE)—Offset 50h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 50h

**Default:** 00000001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:5 | 0x0<br>RW/L | **TCO Base Address (TCOBA):** Provides the 32 bytes of I/O space for TCO logic, mappable anywhere in the 64k I/O space on 32-byte boundaries.<br><br>**Power Well:** Core |
| 4:1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RO | **I/O Space (IOS):** Indicates an I/O Space.<br><br>**Power Well:** Core |

## 59.2.18 TCO Control (TCOCTL)—Offset 54h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 54h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **TCO Base Enable (TCO_BASE_EN):** When set, decode of the I/O range pointed to by the TCO base register is enabled.<br><br>**Power Well:** Core |
| 7:1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/O | **TCO Base Lock (TCO_BASE_LOCK):** When set to 1, this bit locks down the TCO Base Address Register (TCOBASE) at offset 50h.<br>The Base Address Field becomes read-only.<br>This bit becomes locked when a value of 1b is written to it.<br>Writes of 0 to this bit are always ignored.<br>Once locked by writing 1, the only way to clear this bit is to perform a platform reset.<br><br>**Power Well:** Core |

## 59.2.19 Host_Timing (HTIM)—Offset 64h

Setting these timing parameters will affect the SMBus host PHY layer. These values should only be changed when the SMBus is idle.

These timings are based off the 10MHz PHY clock. A higher frequency PHY clock will scale the timings accordingly. This register is in the suspend well.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + 64h

**Default:** 08080000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x8<br>RW | **Thigh:** The time value in this field will be added/subtracted to the Thigh timing parameter as defined in SMB 2.0 spec.<br>• 00000000: -7 Clocks (-700ns)<br>• 00000001: -6 Clocks (-600ns)<br>• ----<br>• 00000111: -1 Clock (-100ns)<br>• 00001000: +0 Clock (0ns)<br>• 00001001: +1 Clock (+100ns)<br>• ----<br>• 11111111: +247 Clocks (+24700ns)<br>***Note:*** These are offsets that control the range of the nominal Thigh value in the PHY layers<br><br>**Power Well:** SUS |
| 23:16 | 0x8<br>RW | **Tiow:** The time value in this field will be added/subtracted to the Tiow timing parameter as defined in SMB 2.0 spec.<br>See Thigh field above<br><br>**Power Well:** SUS |
| 15:12 | 0x0<br>RW | **Thdsta:** The time value in this field will be added/subtracted to the Thdsta timing parameter as defined in SMB 2.0 spec.<br>• 1111: -7 Clocks (-700ns)<br>• 1110: -6 Clocks (-600ns)<br>• ----<br>• x000: +0 Clock (0ns)<br>• 0001: +1 Clock (+100ns)<br>• ----<br>• 0111: +7 Clocks (+700ns)<br><br>**Power Well:** SUS |
| 11:8 | 0x0<br>RW | **Tsusta:** The time value in this field will be added/subtracted to the Tsusta timing parameter as defined in SMB 2.0 spec.<br>See Thdsta field above.<br><br>**Power Well:** SUS |
| 7:4 | 0x0<br>RW | **Tbuf:** The time value in this field will be added/subtracted to the Tbuf timing parameter as defined in SMB 2.0 spec.<br>See Thdsta field above.<br><br>**Power Well:** SUS |
| 3:0 | 0x0<br>RW | **Tsusto:** The time value in this field will be added/subtracted to the Tsusto timing parameter as defined in SMB 2.0 spec.<br>See Thdsta field above.<br><br>**Power Well:** SUS |

## 59.2.20 Manufacturer's ID (MANID)—Offset F8h

This reflects the value of the Manufacturers ID which provides information on the details of the chip revision.

*Note:* A single Manufacturers ID is implemented based on information in the Fuse block and distributed to the devices via a message.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:4] + F8h

**Default:** 01000F1Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 27:24 | 0x1<br>RO/V | **Dot Portion of Process ID (DPID):** Indicates the dot process as x.8.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO/V | **Stepping ID (SID):** This field is incremented for each stepping of the part.<br>***Note:*** This field can be used by software to differentiate steppings when the Revision ID may not change.<br><br>**Power Well:** Core |
| 15:8 | 0xf<br>RO/V | **Manufacturer (MaFa):** 0Fh = Intel.<br><br>**Power Well:** Core |
| 7:0 | 0x1C<br>RO/V | **Process/Dot (PD):** Indicates that the process base is 1271.<br><br>**Power Well:** Core |

## 59.3 Registers in Memory Space—SMBMBAR

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) SMBMBAR.

### 59.3.1 Host Status Register Address (HSTS)—Offset 0h

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no affect.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 0h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C | **BYTE_DONE_STS (BDS):** This bit will be set to 1 when the host controller has received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands) when the 32-byte buffer is not being used.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW/1C | **In Use Status (IUS):** After a full PCI reset, a read to this bit returns a 0.<br>After the first read, subsequent reads will return a 1.<br>A write of a 1 to this bit will reset the next read value to 0.<br>Writing a 0 to this bit has no effect.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW/1C | **SMBALERT_STS (SMSTS):** SoC sets this bit to a 1 to indicates source of the interrupt or SMI# was the SMBAlert# signal.<br>Software resets this bit by writing a 1 to this location.<br>This bit should also be cleared by RSMRST# (but not PLTRST#).<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/1C | **Failed (FAIL):** When set, this indicates that the source of the interrupt or SMI# was a failed bus transaction.<br>This is set in response to the KILL bit being set to terminate the host transaction.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/1C | **Bus Error (BERR):** When set, this indicates the source of the interrupt or SMI# was a transaction collision.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/1C | **Device Error (DERR):** When set, this indicates that the source of the interrupt or SMI# was due one of the following:<br>• Illegal Command Field Unclaimed Cycle (host initiated)<br>• Host Device Time-out Error. CRC Error<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/1C | **Interrupt (INTR):** When set, this indicates that the source of the interrupt or SMI# was the successful completion of its last command.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/1C | **Host Busy (HBSY):** A 1 indicates that the SMBus Host is running a command from the host interface. No SMB registers should be accessed while this bit is set.<br><br>**Power Well:** Core |

## 59.3.2 Host Control Register (HCTL)—Offset 2h

*Note:* A read to this register will clear the pointer in the 32-byte buffer.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 2h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **PEC_EN:** When set to 1, this bit causes the host controller to perform the SMBus transaction with the Packet Error Checking phase appended.<br>For writes, the value of the PEC byte is transferred from the PEC Register.<br>For reads, the PEC byte is loaded in to the PEC Register.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **START:** This write-only bit is used to initiate the command described in the SMB_CMD field.<br>All registers should be setup prior to writing a 1 to this bit position.<br>This bit always reads zero.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **LAST_BYTE:** Used for I2C Read commands as an indication that the next byte will be the last one to be received for that block.<br><br>**Power Well:** Core |
| 4:2 | 0x0<br>RW | **SMB_CMD:** As shown by the bit encoding below, indicates which command the SMBus Host is to perform.<br>If enabled, the SMBus Host will generate an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the SMBus Host will set the device error (DEV_ERR) status bit and generate an interrupt when the START bit is set.<br>The SMBus Controller will perform no command, and will not operate until DEV_ERR is cleared.<br>Val - Command Description:<br>• 000 - Quick: The slave address and read/write value (bit 0) are stored in the tx slave address register.<br>• 001 - Byte: This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command.<br>• 010 - Byte Data: This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register will contain the read data.<br>• 011 - Word Data: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes the DATA0 and DATA1 registers will contain the read data.<br>• 100 - Process Call: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data.<br>• 101 - Block: This command uses the transmit slave address, command, and DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br>• 110 - I2C Read: This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. The SoC will continue reading data until the NAK is received.<br>• 111 - Block-Process: This command uses the transmit slave address, command, DATA0 and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block read, the count is received and stored in the DATA0 register. Bit 0 of the slave address register always indicate a write command. For writes, data is retrieved from the first m (where m is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br>***Note:*** E32B bit in the Auxiliary Control Register must be set for this command to work.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **KILL:** When set, kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#) selected by the SMB_INTRSEL field.<br>This bit, once set, must be cleared to allow the SMB Host Controller to function normally.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **INTREN:** Enable the generation of an interrupt or SMI# upon the completion of the command.<br><br>**Power Well:** Core |

## 59.3.3 Host Command Register (HCMD)—Offset 3h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 3h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Host Command Register (HCMD):** This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.<br><br>**Power Well:** Core |

## 59.3.4 Transmit Slave Address Register (TSA)—Offset 4h

This register is transmitted by the host controller in the slave address field of the SMB protocol. This is the address of the target.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 4h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RW | **ADDRESS (ADDR):** 7-bit address of the targeted slave.<br>***Note:*** Writes to TSA values of A0h - AEh are blocked depending on the setting of the SPD write disable bit in HCFG - HostConfiguration.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **RW:** Direction of the host transfer.<br>• 1 = Read.<br>• 0 = Write.<br><br>**Power Well:** Core |

## 59.3.5 Data 0 Register (HD0)—Offset 5h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 5h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **DATA0/COUNT (DATA0_COUNT):** This field contains the eight bit data sent in the DATA0 field of the SMB protocol.<br>For block write commands, this register reflects the number of bytes to transfer.<br>This register should be programmed to a value between 1 and 32 for block counts.<br>A count of 0 or a count above 32 will result in unpredictable behavior.<br>The host controller does not check or log illegal block counts.<br><br>**Power Well:** Core |

## 59.3.6 Data 1 Register (HD1)—Offset 6h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 6h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **DATA1:** This eight bit register is transmitted in the DATA1 field of the SMB protocol during the execution of any command.<br><br>**Power Well:** Core |

## 59.3.7 Host Block Data (HBD)—Offset 7h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 7h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Block Data (BDTA):** This is either a register, or a pointer into a 32- byte block array, depending upon whether the E32B bit is set in the Auxiliary Control register.<br>When the E32B bit is cleared, this is a register containing a byte of data to be sent on a block write or read from on a block read, just as it behaved on the SoC.<br>When the E32B bit is set, reads and writes to this register are used to access the 32-byte block data storage array. An internal index pointer is used to address the array, which is reset to 0 by reading the HCTL register (offset 02h).<br>The index pointer then increments automatically upon each access to this register.<br>The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0.<br>When the E2B bit is set, for writes, software will write up to 32-bytes to this register as part of the setup for the command. After the Host Controller has sent the Address, Command, and Byte Count fields, it will send the bytes in the SRAM pointed to by this register.<br>When the E2B bit is cleared for writes, software will place a single byte in this register. After the host controller has sent the address, command, and byte count fields, it will send the byte in this register. If there is more data to send, software will write the next series of bytes to the SRAM pointed to by this register and clear the DONE_STS bit.<br>The controller will then send the next byte. During the time between the last byte being transmitted to the next byte being transmitted, the controller will insert wait-states on the interface.<br>When the E2B bit is set for reads, after receiving the byte count into the Data0 register, the first series of data bytes go into the SRAM pointed to by this register. If the byte count has been exhausted or the 32-byte SRAM has been filled, the controller will generate an SMI# or interrupt (depending on configuration) and set the DONE_STS bit.<br>Software will then read the data. During the time between when the last byte is read from the SRAM to when the DONE_STS bit is cleared, the controller will insert wait states on the interface.<br><br>**Power Well:** Core |

## 59.3.8 Packet Error Check Data Register (PEC)—Offset 8h

*Note:* This register may reside in either the core well or the suspend well.

To simplify the implementation, this register will be in the suspend well with the suspend well version of PCI reset (URST33B). This register contains the 8-bit CRC value that is used as the Packet Error Check on SMBus.

For writes, this register is written by software prior to running the command.

For reads, this register is read by software after the read command is completed on SMBus.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [SMBMBAR] + 8h

**Default:** 00h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **PEC_DATA:** This 8-bit register is written with the SMBus PEC data prior to a write transaction. For read transactions, the PEC data is loaded from the SMBus into this register and is then read by software.<br>Software must ensure that the INUSE_STS bit is properly maintained to avoid having this field over-written by a write transaction following a read transaction.<br><br>**Power Well:** Core |

## 59.3.9 Receive Slave Address Register (RSA)—Offset 9h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [SMBMBAR] + 9h

**Default:** 44h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 6:0 | 0x44<br>RW | **SLAVE_ADDR[6:0] (SA_6_0):** This field is the slave address that the SoC decodes for read and write cycles.<br>The default is not 0 so that it can respond even before the CPU comes up (or if the CPU is dead).<br>This register is reset by RSMRST#, but not by PLTRST#.<br><br>**Power Well:** Resume |

## 59.3.10 Slave Data Register (SD)—Offset Ah

**Type:** Memory Mapped I/O Register (Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [SMBMBAR] + Ah

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO/V | **SLAVE_DATA[15:0] (SD_15_0):** This field is the 16-bit data value written by the external SMBus master.<br>The CPU can then read the value from this register.<br>This register is reset by RSMRST#, but not by PLTRST#.<br>• SLAVE_DATA [15:8] corresponds to the Data Message Byte 1 at Slave Write Register 5 in the table.<br>• SLAVE_DATA [7:0] corresponds to the Data Message Byte 0 at Slave Write Register 4 in the table.<br><br>**Power Well:** Resume |

## 59.3.11 Auxiliary Status (AUXS)—Offset Ch

All bits in this register are in the core well.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + Ch

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO/V | **SMT3EN:** Register value determines if SMT3 is enabled and connected on SMLink1.<br>Disabled means the device is not connected to the pins.<br>• 0: Disable.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO/V | **SMT2EN:** Register value determines if SMT2 is enabled and connected on SMLink0.<br>Disabled means the device is not connected to the pins.<br>• 0: Disable.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO/V | **SMT1EN:** Register value determines if SMT1 is enabled and connected on SMBus.<br>Disabled means the device is not connected to the pins.<br>• 0: Disables.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/1C | **CRC Error (CRCE):** This bit is set if a received message contained a CRC error.<br>When this bit is set, the DERR bit of the host status register will also be set.<br>This bit will be set by the controller if a software abort occurs in the middle of the CRC portion of the cycle or an abort happens after the SMBus Controller has received the final data bit transmitted by the external slave.<br><br>**Power Well:** Core |

## 59.3.12 Auxiliary Control (AUXC)—Offset Dh

All bits in this register are in the resume well.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + Dh

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RW | **Enable 32-byte Buffer (E32B):** When set, the Host Block Data register is a pointer into a 32-byte buffer, as opposed to a single register.<br>This enables the block commands to transfer or receive up to 32-bytes before the controller generates an interrupt.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW | **Automatically Append CRC (AAC):** When set, the SMBus Controller will automatically append the CRC.<br>This bit must not be changed during SM Bus transactions, or undetermined behavior will result.<br>It should be programmed only once during the lifetime of the function.<br><br>**Power Well:** Resume |

## 59.3.13 SMLINK_PIN_CTL Register (SMLC)—Offset Eh

*Note:* This register is in the resume well and is reset by RSMRST#.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [SMBMBAR] + Eh

**Default:** 04h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x1<br>RW | **SMLINK_CLK_CTL:**<br>• 0 = SMBus Controller will drive the SMLINK[0] pin low, independent of what the other SMLINK logic would otherwise indicate for the SMLINK(0) pin.<br>• 1 = The SMLINK[0] pin is Not overdriven low. The other SMLINK logic controls the state of the pin.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RO/V | **SMLINK[1]_CUR_STS (SMLINK1_CUR_STS):** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMLINK[1] pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO/V | **SMLINK[0]_CUR_STS (SMLINK0_CUR_STS):** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMLINK[0] pin. It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Core |

## 59.3.14 SMBUS_PIN_CTL Register (SMBC)—Offset Fh

*Note:* This register is in the resume well and is reset by RSMRST#.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 04h

**BAR and Offset:** [SMBMBAR] + Fh

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x1<br>RW | **SMBCLK_CTL:**<br>• 0 = SMBus Controller will drive the SMBCLK pin low, independent of what the other SMB logic would otherwise indicate for the SMBCLK pin.<br>• 1 = The SMBCLK pin is Not overdriven low. The other SMBus logic controls the state of the pin.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RO/V | **SMBDATA_CUR_STS:** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMBDATA pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO/V | **SMBCLK_CUR_STS:** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMBCLK pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |

## 59.3.15 Slave Status Register (SSTS)—Offset 10h

All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [SMBMBAR] + 10h

**Default:** 00h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW/1C | **HOST_NOTIFY_STS (HNS):** The SMBus Controller sets this bit to a 1 when it has completely received a successful Host Notify Command on the SMBus pins.<br>Software reads this bit to determine that the source of the interrupt or SMI# was the reception of the Host Notify Command.<br>Software clears this bit after reading any information needed from the Notify address and data registers by writing a 1 to this bit.<br>***Note:*** The controller will allow the Notify Address and Data registers to be over-written once this bit has been cleared.<br>When this bit is 1, the SMBus Controller will NACK the first byte (host address) of any new Host Notify commands on the SMBus.<br>Writing a 0 to this bit has no effect.<br><br>**Power Well:** Resume |

## 59.3.16 Slave Command Register (SCMD)—Offset 11h

All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally. Also, software must confirm the prior written value before writing to the register again.

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**BAR and Offset:** [SMBMBAR] + 11h

**Default:** 00h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x0<br>RW | **SMBALERT_DIS (SMB_D):** Software sets this bit to 1 to block the generation of the interrupt or SMI# due to the SMBALERT# source.<br>This bit is logically inverted and ANDed with the SMBALERT_STS bit.<br>The resulting signal is distributed to the SMI# and/or interrupt generation logic.<br>This bit does not effect the wake logic.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RW | **HOST_NOTIFY_WKEN (HNW):** Software sets this bit to 1 to enable the reception of a Host Notify command as a wake event.<br>When enabled this event is ORed in with the other SMBus wake events and is reflected in the SMB_WAK_STS bit of the General Purpose Event 0 Status register.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW | **HOST_NOTIFY_INTREN (HNI):** Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS is 1.<br>This enable does not affect the setting of the HOST_NOTIFY_STS bit.<br>When the interrupt is generated, either PIRQB or SMI# is generated, depending on the value of the SMB_SMI_EN bit (D31, F3, Off40h, B1).<br>If the HOST_NOTIFY_STS bit is set when this bit is written to a 1, then the interrupt (or SMI#) will be generated.<br>The interrupt (or SMI#) is logically generated by ANDing the STS and INTREN bits.<br><br>**Power Well:** Resume |

## 59.3.17 Notify Device Address Register (NDA)—Offset 14h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 14h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RO/V | **DEVICE_ADDRESS (Dev_Addr):** This field contains the 7-bit device address received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |

## 59.3.18 Notify Data Low Byte Register (NDLB)—Offset 16h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 16h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **DATA_LOW_BYTE (DLB):** This field contains the first (low) byte of data received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |

## 59.3.19 Notify Data High Byte Register (NDHB)—Offset 17h

**Type:** Memory Mapped I/O Register (Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SMBMBAR] + 17h

**SMBMBAR Type:** PCI Configuration Register (Size: 64 bits)
**SMBMBAR Reference:** [B:0, D:31, F:4] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **DATA_HIGH_BYTE (DHB):** This field contains the second (high) byte of data received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |

## 59.4 Registers in I/O Space—SBA

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) SBA.

### 59.4.1 Host Status Register Address (HSTS)—Offset 0h

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no affect.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 0h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C | **BYTE_DONE_STS (BDS):** This bit will be set to 1 when the host controller has received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands) when the 32-byte buffer is not being used.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW/1C | **In Use Status (IUS):** After a full PCI reset, a read to this bit returns a 0.<br>After the first read, subsequent reads will return a 1.<br>A write of a 1 to this bit will reset the next read value to 0.<br>Writing a 0 to this bit has no effect.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW/1C | **SMBALERT_STS (SMSTS):** The SoC sets this bit to a 1 to indicates source of the interrupt or SMI# was the SMBAlert# signal.<br>Software resets this bit by writing a 1 to this location.<br>This bit should also be cleared by RSMRST# (but not PLTRST#).<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/1C | **Failed (FAIL):** When set, this indicates that the source of the interrupt or SMI# was a failed bus transaction.<br>This is set in response to the KILL bit being set to terminate the host transaction.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/1C | **Bus Error (BERR):** When set, this indicates the source of the interrupt or SMI# was a transaction collision.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/1C | **Device Error (DERR):** When set, this indicates that the source of the interrupt or SMI# was due one of the following:<br>• Illegal Command Field Unclaimed Cycle (host initiated)<br>• Host Device Time-out Error. CRC Error<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/1C | **Interrupt (INTR):** When set, this indicates that the source of the interrupt or SMI# was the successful completion of its last command.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/1C | **Host Busy (HBSY):** A 1 indicates that the SMBus Host is running a command from the host interface. No SMB registers should be accessed while this bit is set.<br><br>**Power Well:** Core |

## 59.4.2 Host Control Register (HCTL)—Offset 2h

*Note:* A read to this register will clear the pointer in the 32-byte buffer.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 2h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **PEC_EN:** When set to 1, this bit causes the host controller to perform the SMBus transaction with the Packet Error Checking phase appended.<br>For writes, the value of the PEC byte is transferred from the PEC Register.<br>For reads, the PEC byte is loaded in to the PEC Register. When this bit is cleared to 0, the SMBus host controller does not perform the transaction with the PEC phase appended.<br>This bit must be written prior to the write in which the START bit is set.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **START:** This write-only bit is used to initiate the command described in the SMB_CMD field.<br>All registers should be setup prior to writing a 1 to this bit position.<br>This bit always reads zero.<br>The HOST_BUSY bit in the Host Status register (offset 00h) can be used to identify when the SMBus Controller has finished the command.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **LAST_BYTE:** Used for I2C Read commands as an indication that the next byte will be the last one to be received for that block.<br>The algorithm and usage model for this bit will be as follows (assume a message of n bytes):<br>• A. When the software sees the BYTE_DONE_STS bit set (bit 7 in the SMBus Host Status Register) for each of bytes 1 through n-2 of the message, the software should then read the Block Data Byte Register to get the byte that was just received.<br>• B. After reading each of bytes 1 to n-2 of the message, the software will then clear the BYTE_DONE_STS bit.<br>• C. After receiving byte n-1 of the message, the software will then set the LAST BYTE bit. The software will then clear the BYTE_DONE_STS bit.<br>• D. The SoC will then receive the last byte of the message (byte n). However, the controller state machine will see the LAST BYTE bit set, and instead of sending an ACK after receiving the last byte, it will instead send a NAK.<br>• E. After receiving the last byte (byte n), the software will still clear the BYTE_DONE_STS bit. However, the LAST_BYTE bit will be irrelevant at that point.<br>***Note:*** This bit may be set when the TCO timer causes the SECOND_TO_STS bit to be set. See the TCO2_STS Register in Volume 1, bit 1 for more details on that bit.<br>The SMBus device driver should clear the LAST_BYTE bit (if it is set) before starting any new command.<br>***Note:*** In addition to I2C Read Commands, the LAST_BYTE bit will also cause Block Read/Write cycles to stop prematurely (at the end of the next byte).<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4:2 | 0x0<br>RW | **SMB_CMD:** As shown by the bit encoding below, indicates which command the SMBus Host is to perform.<br>If enabled, the SoC will generate an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the SoC will set the device error (DEV_ERR) status bit and generate an interrupt when the START bit is set.<br>The controller will perform no command, and will not operate until DEV_ERR is cleared.<br>Val - Command Description:<br>• 000 - Quick: The slave address and read/write value (bit 0) are stored in the tx slave address register.<br>• 001 - Byte: This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command.<br>• 010 - Byte Data: This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register will contain the read data.<br>• 011 - Word Data: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes the DATA0 and DATA1 registers will contain the read data.<br>• 100 - Process Call: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data.<br>• 101 - Block: This command uses the transmit slave address, command, and DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br>• 110 - I2C Read: This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. The SMBus Controller will continue reading data until the NAK is received.<br>• 111 - Block-Process: This command uses the transmit slave address, command, DATA0 and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block read, the count is received and stored in the DATA0 register. Bit 0 of the slave address register always indicate a write command. For writes, data is retrieved from the first m (where m is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register.<br>***Note:*** E32B bit in the Auxiliary Control Register must be set for this command to work.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **KILL:** When set, kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#) selected by the SMB_INTRSEL field.<br>This bit, once set, must be cleared to allow the SMB Host Controller to function normally.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **INTREN:** Enable the generation of an interrupt or SMI# upon the completion of the command.<br><br>**Power Well:** Core |

## 59.4.3 Host Command Register (HCMD)—Offset 3h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 3h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Host Command Register (HCMD):** This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.<br><br>**Power Well:** Core |

## 59.4.4 Transmit Slave Address Register (TSA)—Offset 4h

This register is transmitted by the host controller in the slave address field of the SMB protocol. This is the address of the target.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 4h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RW | **ADDRESS (ADDR):** 7-bit address of the targeted slave.<br>***Note:*** Writes to TSA values of A0h - AEh are blocked depending on the setting of the SPD write disable bit in HCFG - HostConfiguration.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **RW:** Direction of the host transfer.<br>• 1 = Read.<br>• 0 = Write.<br><br>**Power Well:** Core |

## 59.4.5 Data 0 Register (HD0)—Offset 5h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 5h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **DATA0/COUNT (DATA0_COUNT):** This field contains the eight bit data sent in the DATA0 field of the SMB protocol.<br>For block write commands, this register reflects the number of bytes to transfer.<br>This register should be programmed to a value between 1 and 32 for block counts.<br>A count of 0 or a count above 32 will result in unpredictable behavior.<br>The host controller does not check or log illegal block counts.<br><br>**Power Well:** Core |

## 59.4.6 Data 1 Register (HD1)—Offset 6h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 6h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **DATA1:** This eight bit register is transmitted in the DATA1 field of the SMB protocol during the execution of any command.<br><br>**Power Well:** Core |

## 59.4.7 Host Block Data (HBD)—Offset 7h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 7h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Block Data (BDTA):** This is either a register, or a pointer into a 32- byte block array, depending upon whether the E32B bit is set in the Auxiliary Control register.<br>When the E32B bit is cleared, this is a register containing a byte of data to be sent on a block write or read from on a block read, just as it behaved on the SoC.<br>When the E32B bit is set, reads and writes to this register are used to access the 32-byte block data storage array.<br>An internal index pointer is used to address the array, which is reset to 0 by reading the HCTL register (offset 02h).<br>The index pointer then increments automatically upon each access to this register.<br>The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0.<br>When the E2B bit is set, for writes, software will write up to 32-bytes to this register as part of the setup for the command.<br>After the Host Controller has sent the Address, Command, and Byte Count fields, it will send the bytes in the SRAM pointed to by this register.<br>When the E2B bit is cleared for writes, software will place a single byte in this register.<br>After the host controller has sent the address, command, and byte count fields, it will send the byte in this register.<br>If there is more data to send, software will write the next series of bytes to the SRAM pointed to by this register and clear the DONE_STS bit.<br>The controller will then send the next byte.<br>During the time between the last byte being transmitted to the next byte being transmitted, the controller will insert wait-states on the interface.<br>When the E2B bit is set for reads, after receiving the byte count into the Data0 register, the first series of data bytes go into the SRAM pointed to by this register.<br>If the byte count has been exhausted or the 32-byte SRAM has been filled, the controller will generate an SMI# or interrupt (depending on configuration) and set the DONE_STS bit.<br>Software will then read the data. During the time between when the last byte is read from the SRAM to when the DONE_STS bit is cleared, the controller will insert wait states on the interface.<br><br>**Power Well:** Core |

## 59.4.8 Packet Error Check Data Register (PEC)—Offset 8h

*Note:* This register may reside in either the core well or the suspend well.

To simplify the implementation, this register will be in the suspend well with the suspend well version of PCI reset (URST33B). This register contains the 8-bit CRC value that is used as the Packet Error Check on SMBus. For writes, this register is written by software prior to running the command. For reads, this register is read by software after the read command is completed on SMBus.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 8h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **PEC_DATA:** This 8-bit register is written with the SMBus PEC data prior to a write transaction. For read transactions, the PEC data is loaded from the SMBus into this register and is then read by software.<br>Software must ensure that the INUSE_STS bit is properly maintained to avoid having this field over-written by a write transaction following a read transaction.<br><br>**Power Well:** Core |

## 59.4.9 Receive Slave Address Register (RSA)—Offset 9h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 44h

**BAR and Offset:** [SBA] + 9h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 6:0 | 0x44<br>RW | **SLAVE_ADDR[6:0] (SA_6_0):** This field is the slave address that the SMBus Host decodes for read and write cycles.<br>The default is not 0 so that it can respond even before the CPU comes up (or if the CPU is dead).<br>This register is reset by RSMRST#, but not by PLTRST#.<br><br>**Power Well:** Resume |

## 59.4.10 Slave Data Register (SD)—Offset Ah

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [SBA] + Ah

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:0 | 0x0<br>RO/V | **SLAVE_DATA[15:0] (SD_15_0):** This field is the 16-bit data value written by the external SMBus master.<br>The CPU can then read the value from this register.<br>This register is reset by RSMRST#, but not by PLTRST#.<br>• SLAVE_DATA [15:8] corresponds to the Data Message Byte 1 at Slave Write Register 5 in the table.<br>• SLAVE_DATA [7:0] corresponds to the Data Message Byte 0 at Slave Write Register 4 in the table.<br><br>**Power Well:** Resume |

## 59.4.11 Auxiliary Status (AUXS)—Offset Ch

All bits in this register are in the core well.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + Ch

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO/V | **SMT3EN:** Register value determines if SMT3 is enabled and connected on SMLink1.<br>Disabled means the device is not connected to the pins.<br>• 0: Disable.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO/V | **SMT2EN:** Register value determines if SMT2 is enabled and connected on SMLink0.<br>Disabled means the device is not connected to the pins.<br>• 0: Disable.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO/V | **SMT1EN:** Register value determines if SMT1 is enabled and connected on SMBus.<br>Disabled means the device is not connected to the pins.<br>• 0: Disables.<br>• 1: Enable.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/1C | **CRC Error (CRCE):** This bit is set if a received message contained a CRC error.<br>When this bit is set, the DERR bit of the host status register will also be set.<br>This bit will be set by the controller if a software abort occurs in the middle of the CRC portion of the cycle or an abort happens after SMBus Host has received the final data bit transmitted by external slave.<br><br>**Power Well:** Core |

## 59.4.12 Auxiliary Control (AUXC)—Offset Dh

All bits in this register are in the resume well.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + Dh

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RW | **Enable 32-byte Buffer (E32B):** When set, the Host Block Data register is a pointer into a 32-byte buffer, as opposed to a single register.<br>This enables the block commands to transfer or receive up to 32-bytes before the SoC generates an interrupt.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW | **Automatically Append CRC (AAC):** When set, the SoC will automatically append the CRC.<br>This bit must not be changed during SM Bus transactions, or undetermined behavior will result.<br>It should be programmed only once during the lifetime of the function.<br><br>**Power Well:** Resume |

## 59.4.13 SMLINK_PIN_CTL Register (SMLC)—Offset Eh

*Note:* This register is in the resume well and is reset by RSMRST#.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 04h

**BAR and Offset:** [SBA] + Eh

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x1<br>RW | **SMLINK_CLK_CTL:**<br>• 0 = SMBus Controller will drive the SMLINK[0] pin low, independent of what the other SMLINK logic would otherwise indicate for the SMLINK(0) pin.<br>• 1 = The SMLINK[0] pin is Not overdriven low. The other SMLINK logic controls the state of the pin.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RO/V | **SMLINK[1]_CUR_STS (SMLINK1_CUR_STS):** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMLINK[1] pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO/V | **SMLINK[0]_CUR_STS (SMLINK0_CUR_STS):** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMLINK[0] pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Core |

## 59.4.14 SMBUS_PIN_CTL Register (SMBC)—Offset Fh

*Note:* This register is in the resume well and is reset by RSMRST#.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 04h

**BAR and Offset:** [SBA] + Fh

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x1<br>RW | **SMBCLK_CTL:**<br>• 0 = SMBus Controller will drive the SMBCLK pin low, independent of what the other SMB logic would otherwise indicate for the SMBCLK pin.<br>• 1 = The SMBCLK pin is Not overdriven low.<br>The other SMBus logic controls the state of the pin.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RO/V | **SMBDATA_CUR_STS:** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMBDATA pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO/V | **SMBCLK_CUR_STS:** This bit has a default value that is dependent on an external signal level.<br>This returns the value on the SMBCLK pin.<br>It will be 1 to indicate high, 0 to indicate low.<br>This allows software to read the current state of the pin.<br><br>**Power Well:** Resume |

## 59.4.15 Slave Status Register (SSTS)—Offset 10h

All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 10h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW/1C | **HOST_NOTIFY_STS (HNS):** The SMBus Controller sets this bit to a 1 when it has completely received a successful Host Notify Command on the SMBus pins.<br>Software reads this bit to determine that the source of the interrupt or SMI# was the reception of the Host Notify Command.<br>Software clears this bit after reading any information needed from the Notify address and data registers by writing a 1 to this bit.<br>***Note:*** The SMBus Controller will allow the Notify Address and Data registers to be over-written once this bit has been cleared.<br>When this bit is 1, the SMBus Controller will NACK the first byte (host address) of any new Host Notify commands on the SMBus.<br>Writing a 0 to this bit has no effect.<br><br>**Power Well:** Resume |

## 59.4.16 Slave Command Register (SCMD)—Offset 11h

All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally. Also, software must confirm the prior written value before writing to the register again.

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 11h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 2 | 0x0<br>RW | **SMBALERT_DIS (SMB_D):** Software sets this bit to 1 to block the generation of the interrupt or SMI# due to the SMBALERT# source.<br>This bit is logically inverted and ANDed with the SMBALERT_STS bit.<br>The resulting signal is distributed to the SMI# and/or interrupt generation logic.<br>This bit does not effect the wake logic.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RW | **HOST_NOTIFY_WKEN (HNW):** Software sets this bit to 1 to enable the reception of a Host Notify command as a wake event.<br>When enabled this event is ORed in with the other SMBus wake events and is reflected in the SMB_WAK_STS bit of the General Purpose Event 0 Status register.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW | **HOST_NOTIFY_INTREN (HNI):** Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS is 1.<br>This enable does not affect the setting of the HOST_NOTIFY_STS bit.<br>When the interrupt is generated, either PIRQB or SMI# is generated, depending on the value of the SMB_SMI_EN bit (D31, F3, Off40h, B1).<br>If the HOST_NOTIFY_STS bit is set when this bit is written to a 1, then the interrupt (or SMI#) will be generated.<br>The interrupt (or SMI#) is logically generated by ANDing the STS and INTREN bits.<br><br>**Power Well:** Resume |

## 59.4.17 Notify Device Address Register (NDA)—Offset 14h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 14h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:1 | 0x0<br>RO/V | **DEVICE_ADDRESS (Dev_Addr):** This field contains the 7-bit device address received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |

## 59.4.18 Notify Data Low Byte Register (NDLB)—Offset 16h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 16h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **DATA_LOW_BYTE (DLB):** This field contains the first (low) byte of data received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |

## 59.4.19 Notify Data High Byte Register (NDHB)—Offset 17h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [SBA] + 17h

**SBA Type:** PCI Configuration Register (Size: 32 bits)
**SBA Reference:** [B:0, D:31, F:4] + 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RO | **DATA_HIGH_BYTE (DHB):** This field contains the second (high) byte of data received during the Host Notify protocol of the SMBus 2.0 specification.<br>Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.<br><br>**Power Well:** Resume |

## 59.5 Registers in I/O Space—TCOBASE

These registers are accessed in Host I/O Space starting at the relocatable I/O address defined by the Base Address Register (BAR) TCOBASE described in Section 59.2.17, "TCO Base Address (TCOBASE)—Offset 50h" on page 2763.

### 59.5.1 TCO_RLD Register (TRLD)—Offset 0h

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0004h

**BAR and Offset:** [TCOBASE] + 0h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:10 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 9:0 | 0x4<br>RW | **TCORLD:** Reading this register will return the current count of the TCO timer.<br>Writing any value to this register will reload the timer to prevent the timeout.<br><br>**Power Well:** Core |

### 59.5.2 TCO_DAT_IN Register (TDI)—Offset 2h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [TCOBASE] + 2h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **TCO_DAT_IN (TDI):** Data Register for passing commands from the OS to the SMI handler.<br>Writes to this register will cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register.<br><br>**Power Well:** Core |

## 59.5.3 TCO_DAT_OUT Register (TDO)—Offset 3h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 00h

**BAR and Offset:** [TCOBASE] + 3h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **TCO_DAT_OUT (TDO):** Data Register for passing commands from the SMI handler to the OS. Writes to this register will set the TCO_INT_STS bit in the TCO_STS register. It also causes an interrupt, as selected by the TCO_IRQ_SEL bits.<br><br>**Power Well:** Core |

## 59.5.4 TCO1_STS Register (TSTS1)—Offset 4h

Unless otherwise indicated, these bits are sticky and are cleared by writing a 1 to the corresponding bit position.

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [TCOBASE] + 4h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:14 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RO | **TCO Slave Select (TCO_SLVSEL):** This register bit indicates the value of TCO Slave Select Soft Strap.<br>Refer to the Section 19, "Serial Peripheral Interface (SPI)" on page 592 for details.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW/1C | **CPUSERR_STS:** Reserved<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW/1C | **CPUSMI_STS:** Reserved<br><br>**Power Well:** Core |
| 9 | 0x0<br>RW/1C | **CPUSCI_STS:** Reserved<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW/1C | **BIOSWR_STS:** The SoC sets this bit to 1 and generates and SMI# to indicate an illegal attempt to write to the BIOS. (The BIOS write could be either to FirmwareHUB (FWH) or SPI Flash). This occurs when either:<br>• a) The WP bit (10.1.7.4 bit 0) is changed from 0 to 1 and the LE bit (10.1.7.4 bit 1) is also set.<br>• b) Any write is attempted to the BIOS and the WP bit is also set.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C | **NEWCENTURY_STS:** This bit will be set when the year rolls over from 1999 to 2000.<br>If the bit is already 1, it will remain 1.<br>This bit can be cleared either by software writing a 1 back to the bit position, or by RTEST# going active.<br>When this bit is set, an SMI# will be generated. However, this will not be a wake event.<br>***Note:*** This bit 7 is not valid when the RTC battery is first put in (or if the RTC battery does not provide sufficient power when the system is unplugged).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/1C | **TIMEOUT:** Bit set to 1 by the SoC to indicate that the SMI was caused by TCO timer reaching 0.<br>***Note:*** The SMI handler should clear this bit to prevent an immediate re-entry to the SMI handler.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW/1C | **TCO_INT_STS:** Bit set to 1 when SMI handler caused the interrupt by writing to the TCO_DAT_OUT register.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW/1C | **OS_TCO_SMI:** Bit set to 1 when OS code caused an SMI# by writing to the TCO_DAT_IN register.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO/V | **NMI2SMI_STS:** This bit is set when an SMI# occurs because an event occurred that would otherwise have caused an NMI.<br><br>**Power Well:** Core |

## 59.5.5 TCO2_STS Register (TSTS2)—Offset 6h

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [TCOBASE] + 6h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 4 | 0x0<br>RW/1C | **SMLINK_SLAVE_SMI_STS:** Reserved<br><br>**Power Well:** Resume |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Resume |
| 1 | 0x0<br>RW/1C | **SECOND_TO_STS:** The SoC sets this bit to 1 to indicate that the TIMEOUT bit had been (or is currently) set and a second timeout occurred before the TCO_RLD register was written.<br>If this bit is set and the NO_REBOOT config bit is 0, then the SoC will reboot the system after the second timeout.<br>The reboot is done by asserting PLTRST#.<br>This bit is only cleared by writing a 1 to this bit or by a RSMRST#.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RW/1C | **INTRD_DET:** Intruder Detect. Bit set to 1 by the SoC to indicate that an intrusion was detected.<br>This is latched. This bit is cleared by writing a 1 to this bit or by RTEST#.<br>This bit is backed in the RTC Well.<br>***Note:*** If the INTRUDER# signal is active when the software attempts to clear the INTRD_DET bit, the bit will remain as a 1, and the SMI# will be generated again immediately.<br><br>**Power Well:** Resume |

## 59.5.6 TCO1_CNT Register (TCTL1)—Offset 8h

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0000h

**BAR and Offset:** [TCOBASE] + 8h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:13 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RW | **TCO_LOCK:** When set to 1, this bit prevents writes from changing the TCO_EN bit (in offset 30h of Power Management I/O space).<br>Once this bit is set to 1, it can not be cleared by software writing a 0 to this bit location.<br>A core-well reset is required to change this bit from 1 to 0. This bit defaults to 0.<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **TCO_TMR_HALT:** 1 = The TCO timer will halt.<br>It will not count, and thus cannot reach a value that would cause an SMI# or to cause the SECOND_TO_STS bit to be set.<br>This will also prevent rebooting.<br>• 0 = The TCO timer is enabled to count. This is the default.<br><br>**Power Well:** Core |
| 10 | 0x0<br>RW | **SEND_NOW:** Reserved.<br>**Power Well:** Core |
| 9 | 0x0<br>RW | **NMI2SMI_EN:** Setting this bit "1" forces all NMIs to instead cause an SMI#, and will be reported in the TCO1_STS register. This bit is set and the NMI_EN# bit is set to "0", the NMI# will be routed to cause an SMI#. No NMI# will be caused. However, if the GL_SMI_EN bit is not set, then no SMI# will be generated either. If this bit is set but the NMI_EN# bit is set to "1", then no NMI nor SMI# will be generated.<br><br>The following shows the possible combination:<br>• NMI_EN#, GBL_SMI_EN 00: No SMI# based on NMI events ( since no EMI# at all because SMI_EN =0)<br>• NMI_EN#, GBL_SMI_EN 01: SMI# will be caused based on NMI events<br>• NMI_EN#, GBL_SMI_EN 10: No SMI# at all because SMI_EN is 0<br>• NMI_EN#, GBL_SMI_EN 11: No SMI# based on NMI events because NMI_EN#=1<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **NMI_NOW:** Writing a 1 to this bit causes an NMI. this allows the BIOS or SMI handler to force entry to the NMI handler. The NMI handler is expected to clear this bit. Another NMI will not be generated until the bit is cleared by writing a 1 back to the same bit position.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |

## 59.5.7 TCO2_CNT Register (TCTL2)—Offset Ah

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0008h

**BAR and Offset:** [TCOBASE] + Ah

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:6 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Resume |
| 5:4 | 0x0<br>RW | **OS_POLICY:** OS-based software writes to these bits to select the policy that the BIOS will use after the platform resets due the WDT.<br>The following convention is recommended for the BIOS and OS:<br>• 00 Boot normally.<br>• 01 Shut down.<br>• 10 Don’t load OS. Hold in pre-boot state and use LAN to determine next step.<br>• 11 Reserved Implementation.<br>***Note:*** These are just scratch pad bits. They should not be reset when the TCO logic resets the platform due to Watchdog Timer.<br><br>**Power Well:** Resume |
| 3 | 0x1<br>RW | **SMB_ALERT_DISABLE:** Disables GP/SMBALERT# as an alert source for the heartbeats and the SMBus slave.<br>At reset (RSMRST# pin assertion only), this bit is set and the muxed GP/SMBALERT# are disabled.<br><br>**Power Well:** Resume |
| 2:1 | 0x0<br>RW | **INTRD_SEL:** Selects the action to take if the INTRUDER# signal goes active.<br>• 11 Reserved.<br>• 01 Interrupt. For IRQ routing see Section 59.6.1, “TCO Configuration (TCOCFG)—Offset 0h” on page 2801.<br>• 10 SMI#.<br>• 00 INTRUDER# doesn't cause SMI# or interrupt.<br><br>**Power Well:** Resume |
| 0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Resume |

## 59.5.8 LEGACY_ELIM Register (LE)—Offset 10h

**Type:** I/O Register
(Size: 8 bits)

**Default:** 03h

**BAR and Offset:** [TCOBASE] + 10h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x1<br>RW | **IRQ12_CAUSE:** When software sets the bit to 1, IRQ12 will be high (asserted).<br>When software sets the bit to 0, IRQ12 will be low (not asserted).<br>Default for this bit is 1.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RW | **IRQ1_CAUSE:** When software sets the bit to 1, IRQ1 will be high (asserted).<br>When software sets the bit to 0, IRQ1 will be low (not asserted).<br>Default for this bit is 1.<br><br>**Power Well:** Core |

## 59.5.9 TCO_TMR Register (TTMR)—Offset 12h

**Type:** I/O Register
(Size: 16 bits)

**Default:** 0004h

**BAR and Offset:** [TCOBASE] + 12h

**TCOBASE Type:** PCI Configuration Register (Size: 32 bits)
**TCOBASE Reference:** [B:0, D:31, F:4] + 50h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:10 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 9:0 | 0x4<br>RW | **TCOTMR:** Value that is loaded into the timer each time the TCO_RLD register is written.<br>Values of 0000h or 0001h will be ignored and should not be attempted.<br>The timer is clocked at approximately 0.6 seconds, and thus allows timeouts ranging from 1.2 second to 613.8 seconds.<br>***Note:*** The timer has an error of +/- 1 tick (0.6s).<br><br>**Power Well:** Core |

## 59.6 Sideband Registers—Host SMBus Private Configuration

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 59.6.1 TCO Configuration (TCOCFG)—Offset 0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCF] + 0h
**Host Memory Space:** SBREG_BAR + 0xCF0000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RW | **TCO IRQ Enable (IE):** When set, TCO IRQ is enabled, as selected by the TCO_IRQ_SEL field. When cleared, TCO IRQ is disabled.<br><br>**Power Well:** Core |
| 6:3 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **TCO IRQ Select (IS):** Specifies on which IRQ the TCO will internally appear.<br>If not using the APIC, the TCO interrupt must be routed to IRQ9-11, and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts.<br>If using the APIC, the TCO interrupt can also be mapped to IRQ20-23, and can be shared with other interrupts.<br>Bits TCO Map<br>• 000 IRQ9 (maps to 8259 and APIC).<br>• 001 IRQ10 (maps to 8259 and APIC).<br>• 010 IRQ11 (maps to 8259 and APIC).<br>• 011 Reserved 100 IRQ20 (maps to APIC).<br>• 101 IRQ21 (maps to APIC).<br>• 110 IRQ22 (maps to APIC).<br>• 111 IRQ23 (maps to APIC).<br>If the TCO IRQ is programmed to be routed to both the 8259 and the APIC, only one of these interrupt controllers should be enabled to service the interrupt.<br>When setting the these bits, the TCO IRQ Enable (IE) bit must be cleared to prevent glitching.<br>When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC should be programmed for active-high reception.<br>When the interrupt is mapped to APIC interrupts 20 through 23, the APIC should be programmed for active-low reception.<br><br>**Power Well:** Core |

## 59.6.2 General Control (GC)—Offset Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCF] + Ch
**Host Memory Space:** SBREG_BAR + 0xCF0000 + Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:3 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **SMB Static Clock Gating Enable (SMBSCGE):** SMB cluster static clock gating is enabled.<br>***Note:*** Setting this bit will also clock gate all the TCO logic functionality.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **No Reboot (NR):** This bit is set when the "No TCO Reboot Enable" SoC Hard Pin Strap pin (pin name GPIO[12]) is sampled high on the rising edge of the active-low RSMRST_N SoC signal pin.<br><br>This bit may be set or cleared by software if the Hard Pin Strap is sampled low but may not override the strap when it indicates No TCO Reboot.<br><br>When set, the TCO Watchdog Timer counts down and generates the SMI# on the first timeout, but will not reboot on the second timeout.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Function Disable (FD):** When set to one, this disables the PCI configuration register space for the SMBus device.<br><br>**Power Well:** Core |

## 59.6.3 Power Control Enable (PCE)—Offset 10h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCF] + 10h
**Host Memory Space:** SBREG_BAR + 0xCF0000 + 10h

**Default:** 00000009h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **Hardware Autonomous Enable (HAE):** When set, the SMB will automatically engage power gating when it has reached its idle condition.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x1<br>RO | **Sleep Enable (SE):** When this bit is clear, the SMB will never assert Sleep to the retention flops. If set, then SMB may assert Sleep during power gating.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RO | **D3-Hot Enable (D3HE):** No support for D3 Hot power gating.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO | **I3 Enable (I3E):** No support for S0i3 power gating.<br><br>**Power Well:** Core |
| 0 | 0x1<br>RW | **PMC Request Enable (PMCRE):** When set to 1, the SMB will engage power gating if it is idle and the power-gating request is granted by the PMC.<br><br>**Power Well:** Core |

## 59.7 Sideband Registers—CLTT SMBus Configuration Control

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 59.7.1 CLTT Status (CLTT_STS)—Offset 0h

CLTT Status

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 0h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO/V | **Done (DONE):** Indicates that HW has completed the SMBusCmd.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO/V | **Busy (BUSY):** Indicates that HW is busy completing an SMBusCmd.<br><br>**Power Well:** Core |

## 59.7.2 Ch0 Dimm0 Cfg Status (C0D0CFGSTS)—Offset 10h

Memory Channel 0 DIMM 0 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 10h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 10h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.3 Ch0 Dimm1 Cfg Status (C0D1CFGSTS)—Offset 14h

Memory Channel 0 DIMM 1 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 14h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 14h

**Default:** 01000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x1<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.4 Ch0 Dimm2 Cfg Status (C0D2CFGSTS)—Offset 18h

Memory Channel 0 DIMM 2 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 18h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 18h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.5 Ch0 Dimm3 Cfg Status (C0D3CFGSTS)—Offset 1Ch

Memory Channel 0 DIMM 3 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 1Ch
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 1Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.6 Ch1 Dimm0 Cfg Status (C1D0CFGSTS)—Offset 20h

Memory Channel 1 DIMM 0 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 20h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 20h

**Default:** 02000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x2<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.7 Ch1 Dimm1 Cfg Status (C1D1CFGSTS)—Offset 24h

Memory Channel 1 DIMM 1 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 24h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 24h

**Default:** 03000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x3<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.8 Ch1 Dimm2 Cfg Status (C1D2CFGSTS)—Offset 28h

Memory Channel 1 DIMM 2 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 28h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.9 Ch1 Dimm3 Cfg Status (C1D3CFGSTS)—Offset 2Ch

Memory Channel 1 DIMM 3 Configuration and Status Register.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 2Ch
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Valid (VALID):** Channel DIMM is Populated.<br><br>**Power Well:** Core |
| 30:24 | 0x0<br>RW | **Slave ID (SLAVE_ID):** SMBus Slave ID.<br><br>**Power Well:** Core |
| 23:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO/V | **Read_Data (READ_DATA):** Data retrieved from SMBus.<br><br>**Power Well:** Core |

## 59.7.10 Ch0 Dimm Temp Status (C0TEMPSTS0)—Offset 30h

Memory Channel 0 DIMM TSOD Temperature Status Register. This register is protected by SAI WAC #1 and can be locked by CLTT_SMBUS_Mode (CSMBM) register Bit[1].

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 30h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/V/L | **Read_Temp_Data1 (READ_TEMP_DATA1):** Read_Temp_Data1: Data retrieved from DIMM1.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW/V/L | **Read_Temp_Data0 (READ_TEMP_DATA0):** Read_Temp_Data0: Data retrieved from DIMM0.<br><br>**Power Well:** Core |

## 59.7.11 Ch0 Dimm Temp Status (C0TEMPSTS1)—Offset 34h

Memory Channel 0 DIMM TSOD Temperature Status Register. This register is protected by SAI WAC #1 and can be locked by CLTT_SMBUS_Mode (CSMBM) register Bit[1].

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 34h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 34h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/V/L | **Read_Temp_Data3 (READ_TEMP_DATA3):** Read_Temp_Data3: Data retrieved from DIMM3.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW/V/L | **Read_Temp_Data2 (READ_TEMP_DATA2):** Read_Temp_Data2: Data retrieved from DIMM2.<br><br>**Power Well:** Core |

## 59.7.12 Ch1 Dimm Temp Status (C1TEMPSTS0)—Offset 38h

Memory Channel 1 DIMM TSOD Temperature Status Register. This register is protected by SAI WAC #1 and can be locked by CLTT_SMBUS_Mode (CSMBM) register Bit[1].

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 38h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 38h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/V/L | **Read_Temp_Data1 (READ_TEMP_DATA1):** Read_Temp_Data1: Data retrieved from DIMM1.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW/V/L | **Read_Temp_Data0 (READ_TEMP_DATA0):** Read_Temp_Data0: Data retrieved from DIMM0.<br><br>**Power Well:** Core |

## 59.7.13 Ch1 Dimm Temp Status (C1TEMPSTS1)—Offset 3Ch

Memory Channel 1 DIMM TSOD Temperature Status Register. This register is protected by SAI WAC #1 and can be locked by CLTT_SMBUS_Mode (CSMBM) register Bit[1].

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 3Ch
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/V/L | **Read_Temp_Data3 (READ_TEMP_DATA3):** Read_Temp_Data3: Data retrieved from DIMM3.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RW/V/L | **Read_Temp_Data2 (READ_TEMP_DATA2):** Read_Temp_Data2: Data retrieved from DIMM2.<br><br>**Power Well:** Core |

## 59.7.14 Host Configuration (HCFG)—Offset 40h

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xCC] + 40h
**Host Memory Space:** SBREG_BAR + 0xCC0000 + 40h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/O | **SPD Write Disable (SPD_WRITE_DISABLE):** Reserved.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/V | **SSRESET:** Soft SMBUS Reset: When this bit is 1, the SMBus state machine and logic is reset. The HW will reset this bit to 0 when reset operation is completed.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **I2C_EN:** When this bit is 1, the SMB Controller is enabled to communicate with I2C devices. This will change the formatting of some commands. When this bit is 0, behavior is for SMBus.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **SMB_SMI_EN:** When this bit is set, any source of an SMB interrupt will instead be routed to generate an SMI#. Refer to Section 6.7 (Interrupts/SMI#) for details.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/V | **HST_EN:** When set, the SMB Host Controller interface is enabled to execute commands. The HST_INT_EN bit needs to be enabled in order for the SMB Host Controller to interrupt or SMI#. Additionally, the SMB Host Controller will not respond to any new requests until all interrupt requests have been cleared.<br><br>**Power Well:** Core |

§ §

# 60 SPI Controller - B0, D31, F5

## 60.1 Introduction and Index

The host-accessible registers for the SPI Controller are described here.

To determine the number of devices and functions available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

### 60.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 5. The offset addresses and default values are listed.

**Table 60-1. Summary of PCI Configuration Registers—0/31/5**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 19E08086 | "Device ID and Vendor ID (BIOS_SPI_DID_VID)—Offset 0h" on page 2815 |
| 4 | 4 | 00000400 | "Status and Command (BIOS_SPI_STS_CMD)—Offset 4h" on page 2816 |
| 8 | 4 | 0C800000 | "Revision ID (BIOS_SPI_CC_RID)—Offset 8h" on page 2818 |
| C | 4 | 00000000 | "BIST, Header Type, Latency Timer, Cache Line Size (BIOS_SPI_BIST_HTYPE_LT_CLS)—Offset Ch" on page 2818 |
| 10 | 4 | 00000000 | "SPI BAR0 MMIO (BIOS_SPI_BAR0)—Offset 10h" on page 2819 |
| 28 | 4 | 00000000 | "Cardbus CIS Pointer (BIOS_SPI_CCP)—Offset 28h" on page 2819 |
| 2C | 4 | 00000000 | "Subsystem and Vendor ID (BIOS_SPI_SID_SVID)—Offset 2Ch" on page 2819 |
| 30 | 4 | 00000000 | "Expansion ROM Base Address (BIOS_SPI_XRBAR)—Offset 30h" on page 2820 |
| 34 | 4 | 00000000 | "Capabilities List Pointer (BIOS_SPI_CAPP)—Offset 34h" on page 2820 |
| D0 | 4 | 00000000 | "SPI Unsupported Request Status (BIOS_SPI_UR_STS_CTL)—Offset D0h" on page 2820 |
| D8 | 4 | 0000FFCF | "BIOS Decode Enable (BIOS_SPI_BDE)—Offset D8h" on page 2821 |
| DC | 4 | 00000028 | "BIOS Control (BIOS_SPI_BC)—Offset DCh" on page 2823 |

## 60.1.2 Host Memory Space—BIOS_SPI_BAR0

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 60-2. Summary of Memory Mapped I/O Registers—BIOS_SPI_BAR0**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 4 | 00000000 | "SPI BIOS MMIO PRI (BIOS_BFPREG)—Offset 0h" on page 2825 |
| 4 | 4 | 00002000 | "Hardware Sequencing Flash Status and Control (BIOS_HSFSTS_CTL)—Offset 4h" on page 2826 |
| 8 | 4 | 00000000 | "Flash Address (BIOS_FADDR)—Offset 8h" on page 2828 |
| C | 4 | 00000000 | "Discrete Lock Bits (BIOS_DLOCK)—Offset Ch" on page 2829 |
| 10 | 4 | 00000000 | "Flash Data 0 (BIOS_FDATA0)—Offset 10h" on page 2830 |
| 14 | 4 | 00000000 | "Flash Data 1 (BIOS_FDATA1)—Offset 14h" on page 2830 |
| 18 | 4 | 00000000 | "Flash Data 2 (BIOS_FDATA2)—Offset 18h" on page 2830 |
| 1C | 4 | 00000000 | "Flash Data 3 (BIOS_FDATA3)—Offset 1Ch" on page 2831 |
| 20 | 4 | 00000000 | "Flash Data 4 (BIOS_FDATA4)—Offset 20h" on page 2831 |
| 24 | 4 | 00000000 | "Flash Data 5 (BIOS_FDATA5)—Offset 24h" on page 2831 |
| 28 | 4 | 00000000 | "Flash Data 6 (BIOS_FDATA6)—Offset 28h" on page 2832 |
| 2C | 4 | 00000000 | "Flash Data 7 (BIOS_FDATA7)—Offset 2Ch" on page 2832 |
| 30 | 4 | 00000000 | "Flash Data 8 (BIOS_FDATA8)—Offset 30h" on page 2832 |
| 34 | 4 | 00000000 | "Flash Data 9 (BIOS_FDATA9)—Offset 34h" on page 2833 |
| 38 | 4 | 00000000 | "Flash Data 10 (BIOS_FDATA10)—Offset 38h" on page 2833 |
| 3C | 4 | 00000000 | "Flash Data 11 (BIOS_FDATA11)—Offset 3Ch" on page 2833 |
| 40 | 4 | 00000000 | "Flash Data 12 (BIOS_FDATA12)—Offset 40h" on page 2834 |
| 44 | 4 | 00000000 | "Flash Data 13 (BIOS_FDATA13)—Offset 44h" on page 2834 |
| 48 | 4 | 00000000 | "Flash Data 14 (BIOS_FDATA14)—Offset 48h" on page 2834 |
| 4C | 4 | 00000000 | "Flash Data 15 (BIOS_FDATA15)—Offset 4Ch" on page 2834 |
| 50 | 4 | 000042C2 | "Flash Region Access Permissions (BIOS_FRACC)—Offset 50h" on page 2835 |
| 54 | 4 | 00007FFF | "Flash Region 0 (BIOS_FREG0)—Offset 54h" on page 2836 |
| 58 | 4 | 00007FFF | "Flash Region 1 (BIOS_FREG1)—Offset 58h" on page 2836 |
| 5C | 4 | 00007FFF | "Flash Region 2 (BIOS_FREG2)—Offset 5Ch" on page 2837 |
| 60 | 4 | 00007FFF | "Flash Region 3 (BIOS_FREG3)—Offset 60h" on page 2837 |
| 64 | 4 | 00007FFF | "Flash Region 4 (BIOS_FREG4)—Offset 64h" on page 2838 |
| 68 | 4 | 00007FFF | "Flash Region 5 (BIOS_FREG5)—Offset 68h" on page 2838 |
| 6C | 4 | 00007FFF | "Flash Region 6 (BIOS_FREG6)—Offset 6Ch" on page 2839 |
| 70 | 4 | 00007FFF | "Flash Region 7 (BIOS_FREG7)—Offset 70h" on page 2839 |
| 74 | 4 | 00007FFF | "Flash Region 8 (BIOS_FREG8)—Offset 74h" on page 2840 |
| 78 | 4 | 00007FFF | "Flash Region 9 (BIOS_FREG9)—Offset 78h" on page 2840 |
| 7C | 4 | 00000000 | "Flash Region 10 (BIOS_FREG10)—Offset 7Ch" on page 2841 |
| 80 | 4 | 00000000 | "Flash Region 11 (BIOS_FREG11)—Offset 80h" on page 2841 |
| 84 | 4 | 00000000 | "Flash Protected Range 0 (BIOS_FPR0)—Offset 84h" on page 2842 |
| 88 | 4 | 00000000 | "Flash Protected Range 1 (BIOS_FPR1)—Offset 88h" on page 2843 |
| 8C | 4 | 00000000 | "Flash Protected Range 2 (BIOS_FPR2)—Offset 8Ch" on page 2844 |

## Table 60-2. Summary of Memory Mapped I/O Registers—BIOS_SPI_BAR0 (Continued)

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 90 | 4 | 00000000 | "Flash Protected Range 3 (BIOS_FPR3)—Offset 90h" on page 2845 |
| 94 | 4 | 00000000 | "Flash Protected Range 4 (BIOS_FPR4)—Offset 94h" on page 2846 |
| 98 | 4 | 00000000 | "Global Protected Range 0 (BIOS_GPR0)—Offset 98h" on page 2847 |
| A0 | 4 | FE000000 | "Software Sequencing Flash Status and Control (BIOS_SSFSTS_CTL)—Offset A0h" on page 2848 |
| A4 | 4 | 00000000 | "Prefix Opcode and Opcode Type Configuration (BIOS_PREOP_OPTYPE)—Offset A4h" on page 2850 |
| A8 | 4 | 00000000 | "Opcode Menu0 Configuration (BIOS_OPMENU0)—Offset A8h" on page 2851 |
| AC | 4 | 00000000 | "Opcode Menu1 Configuration (BIOS_OPMENU1)—Offset ACh" on page 2852 |
| B0 | 4 | 00000000 | "Secondary Flash Region Access Permissions (BIOS_SFRACC)—Offset B0h" on page 2852 |
| B4 | 4 | 00000000 | "Flash Descriptor Observability Control (BIOS_FDOC)—Offset B4h" on page 2853 |
| B8 | 4 | 00000000 | "Flash Descriptor Observability Data (BIOS_FDOD)—Offset B8h" on page 2854 |
| C0 | 4 | 00000000 | "Additional Flash Control (BIOS_AFC)—Offset C0h" on page 2854 |
| C4 | 4 | 00002000 | "Vendor Specific Component Capabilities for Component 0 (BIOS_SFDP0_VSCC0)—Offset C4h" on page 2855 |
| C8 | 4 | 00002000 | "Vendor Specific Component Capabilities for Component 1 (BIOS_SFDP1_VSCC1)—Offset C8h" on page 2858 |
| CC | 4 | 00000000 | "Parameter Table Index (BIOS_PTINX)—Offset CCh" on page 2860 |
| D0 | 4 | 00000000 | "Parameter Table Data (BIOS_PTDATA)—Offset D0h" on page 2861 |
| D4 | 4 | 00000000 | "SPI Bus Requester Status (BIOS_SBRS)—Offset D4h" on page 2862 |
| E0 | 4 | 00000000 | "Flash Region 12 (BIOS_FREG12)—Offset E0h" on page 2863 |
| E4 | 4 | 00000000 | "Flash Region 13 (BIOS_FREG13)—Offset E4h" on page 2863 |
| E8 | 4 | 00000000 | "Flash Region 14 (BIOS_FREG14)—Offset E8h" on page 2864 |
| EC | 4 | 00000000 | "Flash Region 15 (BIOS_FREG15)—Offset ECh" on page 2864 |
| F0 | 4 | 00000000 | "Set Strap Msg Lock (SSML)—Offset F0h" on page 2865 |
| F4 | 4 | 00000000 | "Set Strap Msg Control (SSMC)—Offset F4h" on page 2865 |
| F8 | 4 | 00000000 | "Set Strap Msg Data (SSMD)—Offset F8h" on page 2866 |
| 108 | 4 | 00000000 | "RPMC SFDP Table (BIOS_RPMC0_D0)—Offset 108h" on page 2866 |
| 10C | 4 | 00000000 | "RPMC SFDP Table (BIOS_RPMC1_D0)—Offset 10Ch" on page 2867 |
| 110 | 4 | 00000000 | "RPMC SFDP Table (BIOS_RPMC0_D1)—Offset 110h" on page 2867 |
| 114 | 4 | 00000000 | "RPMC SFDP Table (BIOS_RPMC1_D1)—Offset 114h" on page 2867 |
| 118 | 4 | 000000C2 | "BIOS Master Read Access Permissions (BIOS_BM_RAP)—Offset 118h" on page 2868 |
| 11C | 4 | 00000042 | "BIOS Master Write Access Permissions (BIOS_BM_WAP)—Offset 11Ch" on page 2868 |
| 1B0 | 4 | 00000000 | "IE Flash Protected Range 0 (BIOS_IE_PR0)—Offset 1B0h" on page 2869 |
| 1B4 | 4 | 00000000 | "IE Flash Protected Range 1 (BIOS_IE_PR1)—Offset 1B4h" on page 2870 |
| 1B8 | 4 | 00000000 | "IE Flash Protected Range 2 (BIOS_IE_PR2)—Offset 1B8h" on page 2871 |
| 1BC | 4 | 00000000 | "IE Flash Protected Range 3 (BIOS_IE_PR3)—Offset 1BCh" on page 2872 |
| 1C0 | 4 | 00000000 | "IE Flash Protected Range 4 (BIOS_IE_PR4)—Offset 1C0h" on page 2873 |

## 60.2 Registers in Configuration Space

### 60.2.1 Device ID and Vendor ID (BIOS_SPI_DID_VID)—Offset 0h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 0h

**Default:** 19E08086h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x19e0<br>RO/V | **Device Identification (DID):** |
| 15:0 | 0x8086<br>RO | **Vendor Identification (VID):** This field identifies the manufacturer of the device. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness.<br>0x8086 indicates Intel. |

## 60.2.2 Status and Command (BIOS_SPI_STS_CMD)—Offset 4h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 4h

**Default:** 00000400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C/V | **Detected Parity Error (DPE):** Detected Parity Error (DPE). |
| 30 | 0x0<br>RW/1C/V | **Signaled System Error (SSE):** Signaled System Error (SSE). |
| 29 | 0x0<br>RO | **Received Master Abort (RMA):** Received Master Abort (RMA). |
| 28 | 0x0<br>RO | **Received Target Abort (RTA):** Received Target Abort (RTA). |
| 27 | 0x0<br>RW/1C/V | **Signaled Target Abort (STA):** Signaled Target Abort (STA). |
| 26:25 | 0x0<br>RO | **Devsel Timing (DEVT):** Devsel Timing (DEVT). |
| 24 | 0x0<br>RO | **Master Data Parity Error (MDPE):** Master Data Parity Error (MDPE). |
| 23 | 0x0<br>RO | **Fast Back to Back Capable (FBTBC):** Fast Back to Back Capable (FBTBC). |
| 22 | 0x0<br>RO | **Reserved (RSVD0):** Reserved (RSVD). |
| 21 | 0x0<br>RO | **66 MHz Capable (MCAP):** Not 66MHz Capable Device. |
| 20 | 0x0<br>RO | **Capabilities List (CAPL):** Capabilities List. |
| 19 | 0x0<br>RO | **Interrupt Status (INTS):** Interrupt Status. |
| 18:11 | 0x0<br>RO | **Reserved (RSVD1):** Reserved. |
| 10 | 0x1<br>RO | **Interrupt Disable (INTD):** Interrupt Disable. |
| 9 | 0x0<br>RO | **Fast Back to Back Enable (FBTBEN):** Fast Back to Back Enable. |
| 8 | 0x0<br>RW | **System Error Enable (SERREN):** System Error Enable. |
| 7 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 6 | 0x0<br>RW | **Parity Error Response (PERRR):** Parity Error Response. |
| 5 | 0x0<br>RO | **VGA Palette Snoop (VGAPS):** VGA Palette Snoop. |
| 4 | 0x0<br>RO | **Memory Write and Invalidate Enable (MWRIEN):** Memory Write and Invalidate Enable. |
| 3 | 0x0<br>RO | **Special Cycles (SPCYC):** Special Cycles. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **Bus Master Enable (BME):** Bus Master Enable. |
| 1 | 0x0<br>RW | **Memory Space Enable (MSE):** Memory Space Enable. |
| 0 | 0x0<br>RO | **IO Space Enable (IOSE):** IO Space Enable. |

## 60.2.3 Revision ID (BIOS_SPI_CC_RID)—Offset 8h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 8h

**Default:** 0C800000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0xc<br>RO/V | **Base Class Code (BCC):** Base Class Code. |
| 23:16 | 0x80<br>RO/V | **Sub-Class Code (SCC):** Sub-Class Code. |
| 15:8 | 0x0<br>RO/V | **Programming Interface (PI):** Programming Interface. |
| 7:0 | 0x0<br>RO/V | **Revision ID (RID):** Indicates the part revision. |

## 60.2.4 BIST, Header Type, Latency Timer, Cache Line Size (BIOS_SPI_BIST_HTYPE_LT_CLS)—Offset Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 23 | 0x0<br>RO/V | **Multi-function Device (MFD):** Multi-function Device. |
| 22:16 | 0x0<br>RO | **Header Type (HTYPE):** Header Type. |
| 15:8 | 0x0<br>RO | **Latency Timer (LT):** Latency Timer. |
| 7:0 | 0x0<br>RO | **Cacheline Size (CLSZ):** Cacheline Size. |

## 60.2.5 SPI BAR0 MMIO (BIOS_SPI_BAR0)—Offset 10h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 10h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RW | **Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region.<br>The Host/BIOS MMIO registers in the flash controller are offset from this BAR. |
| 11:4 | 0x0<br>RO | **Memory Size (MEMSIZE):** Hardwired to 0 to indicate 4KB of memory space. |
| 3 | 0x0<br>RO | **Prefetchable (PREFETCH):** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.<br>Hardwired to 1 to indicate the device's memory space as prefetchable. |
| 2:1 | 0x0<br>RO | **Type (TYP):** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space. |
| 0 | 0x0<br>RO | **Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR. |

## 60.2.6 Cardbus CIS Pointer (BIOS_SPI_CCP)—Offset 28h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 60.2.7 Subsystem and Vendor ID (BIOS_SPI_SID_SVID)—Offset 2Ch

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RW/O | **Subsystem ID (SSID):** Subsystem ID. |
| 15:0 | 0x0<br>RW/O | **Subsystem Vendor ID (SSVID):** Subsystem Vendor ID. |

## 60.2.8 Expansion ROM Base Address (BIOS_SPI_XRBAR)—Offset 30h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 60.2.9 Capabilities List Pointer (BIOS_SPI_CAPP)—Offset 34h

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + 34h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 7:0 | 0x0<br>RO | **Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list. |

## 60.2.10 SPI Unsupported Request Status (BIOS_SPI_UR_STS_CTL)—Offset D0h

See the IOSF Specification for required behavior.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 1 | 0x0<br>RW/1C/V | **Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unsupported Request that is not considered an Advisory Non-Fatal error.<br>Cleared to 0 when software writes a 1 to this register. |
| 0 | 0x0<br>RW | **Unsupported Request Reporting Enabled (URRE):** If set to 1 by software, the flash controller generates a DoSERR sideband message when the URD bit transitions from 0 to 1. |

## 60.2.11 BIOS Decode Enable (BIOS_SPI_BDE)—Offset D8h

This register only effects BIOS decode if BIOS is resident on SPI.

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + D8h

**Default:** 0000FFCFh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** |
| 15 | 0x1<br>RO | **F8-FF Enable (EF8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFF80000h - FFFFFFFFh |
| 14 | 0x1<br>RW | **F0-F8 Enable (EF0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFF00000h - FFF7FFFFh |
| 13 | 0x1<br>RW | **E8-EF Enable (EE8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFE80000h - FFEFFFFFh |
| 12 | 0x1<br>RW | **E0-E8 Enable (EE0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFE00000h - FFE7FFFFh |
| 11 | 0x1<br>RW | **D8-DF Enable (ED8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFD80000h - FFDFFFFFh |
| 10 | 0x1<br>RW | **D0-D7 Enable (ED0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFD00000h - FFD7FFFFh |
| 9 | 0x1<br>RW | **C8-CF Enable (EC8):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFC80000h - FFCFFFFFh |
| 8 | 0x1<br>RW | **C0-C7 Enable (EC0):** Enables decoding of 512K of the following BIOS range:<br>• Data space: FFC00000h - FFC7FFFFh |
| 7 | 0x1<br>RW | **Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h - FFFFFh<br>***Note:*** Decode for the BIOS legacy F segment is enabled by the LFE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit. |
| 6 | 0x1<br>RW | **Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h - EFFFFh<br>***Note:*** Decode for the BIOS legacy E segment is enabled by the LEE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit. |
| 5:4 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 3 | 0x1<br>RW | **70-7F Enable (E70):** Enables decoding of 1 MB of the following BIOS range:<br>• Data space: FF700000h - FF7FFFFFh |
| 2 | 0x1<br>RW | **60-6F Enable (E60):** Enables decoding of 1 MB of the following BIOS range:<br>• Data space: FF600000h - FF6FFFFFh |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x1<br>RW | **50-5F Enable (E50):** Enables decoding of 1 MB of the following BIOS range:<br>• Data space: FF500000h - FF5FFFFFh |
| 0 | 0x1<br>RW | **40-4F Enable (E40):** Enables decoding of 1 MB of the following BIOS range:<br>• Data space: FF400000h - FF4FFFFFh |

## 60.2.12 BIOS Control (BIOS_SPI_BC)—Offset DCh

**Type:** PCI Configuration Register (Size: 32 bits)

**Offset:** [B:0, D:31, F:5] + DCh

**Default:** 00000028h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 11 | 0x0<br>RW/L | **Async SMI Enable for BIOS Write Protection (ASE_BWP):** When set to '1' the flash controller will generate an SMI when it blocks a BIOS write or erase due to WPD=0.<br>The value in this field can be written by software as long as the BIOS Interface Lock-Down (BILD) is not set. |
| 10 | 0x0<br>RO/V | **Asynchronous SMI Status (SPI_ASYNC_SS):** Status indication that the SPI Flash Controller has asserted an asynchronous SMI.<br>Hardware clears the bit when it sends the De-assert SMI message.<br>• 0 = Default state<br>• 1 = SPI flash controller asserted asynchronous SMI |
| 9 | 0x0<br>RW/L | Reserved. |
| 8 | 0x0<br>RW/1C/V | **Synchronous SMI Status (SPI_SYNC_SS):** Status indication that the SPI Flash Controller has asserted a synchronous SMI.<br>Hardware clears the bit when it sends the De-assert Synchronous SMI message.<br>• 0 = Default state<br>• 1 = SPI flash controller asserted Synchronous SMI |
| 7 | 0x0<br>RW/L | **BIOS Interface Lock-Down (BILD):** When set, prevents TS and BBS from being changed. This bit can only be written from 0 to 1 once. |
| 6 | 0x0<br>RW/V/L | **Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. For the default, “Functional Strap” section of Signal Description chapter for details.<br>Bit Description<br>• 0 = SPI<br>• 1 = LPC<br>The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down (BILD) is not set |
| 5 | 0x1<br>RW/L | **Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until the CPU sets the InSMM.STS bit.<br>If this bit [5] is set, then WPD must be a '1' and InSMM.STS must be '1' also in order to write to BIOS region of SPI Flash.<br>If this bit [5] is clear, then the InSMM.STS is a don't care.<br>This bit is locked by LE. |
| 4 | 0x0<br>RO/V | **Top Swap Status (TSS):** This bit provides a read-only path to view the state of the Top Swap bit. It is duplicated here to be consistent with the LPC version of the BC register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3:2 | 0x2<br>RW | **SPI Read Configuration (SRC):** These bits are located in PCI Config space to allow them to be set early in the boot flow. This 2-bit field controls two policies related to BIOS reads on the SPI interface:<br>Bit 3- Prefetch Enable<br>Bit 2- Cache Disable<br>Settings are summarized below:<br>Bits 3:2 Description<br>• 00 No prefetching, but caching enabled. Direct Memory reads load the read buffer cache with "valid" data, allowing repeated reads to the same range to complete quickly<br>• 01 No prefetching and no caching. One-to-one correspondence of host BIOS reads to SPI cycles. This value can be used to invalidate the cache.<br>• 10 Prefetching and Caching enabled. This mode is used for long sequences of short reads to consecutive addresses (i.e. shadowing).<br>• 11 Illegal. Caching must be enabled when Prefetching is enabled. This eliminates the need for a complex prefetch-flushing mechanism. |
| 1 | 0x0<br>RW/L | **Lock Enable (LE):** When set, setting the WPD bit will cause SMI. When cleared, setting the WPD bit will not cause SMI.<br>Once set, this bit can only be cleared by a PLTRST#.<br>When this bit is set, EISS - bit [5] of this register is locked down. |
| 0 | 0x0<br>RW | **Write Protect Disable (WPD):** When set, access to the BIOS space is enabled for both read and write cycles to BIOS.<br>When cleared, only read cycles are permitted to the FWH or SPI flash.<br>When this bit is written from a '0' to a '1' and the LE bit is also set, an SMI# is generated.<br>This ensures that only SMM code can update BIOS. |

## 60.3 Registers in Memory Space—BIOS_SPI_BAR0

These registers are accessed in Host Memory Space starting at the relocatable memory address defined by the Base Address Register (BAR) BIOS_SPI_BAR0.

### 60.3.1 SPI BIOS MMIO PRI (BIOS_BFPREG)—Offset 0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 0h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Shadowed BIOS Region Select (SBRS):** |
| 30:16 | 0x0<br>RO/V | **BIOS Flash Primary Region Limit (PRL):** This specifies address bits 26:12 for the Primary Region Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Limit, or the Flash Descriptor.FLREG6.Region Limit depending on the BFPREG.SBRS bit. |
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:0 | 0x0<br>RO/V | **BIOS Flash Primary Region Base (PRB):** This specifies address bits 26:12 for the Primary Region Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Base, or the Flash Descriptor.FLREG6.Region Base depending on the BFPREG.SBRS bit. |

## 60.3.2 Hardware Sequencing Flash Status and Control (BIOS_HSFSTS_CTL)—Offset 4h

Note: If operating in Non-Descriptor mode, hardware sequencing is not allowed.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 4h

**Default:** 00002000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Flash SPI SMI# Enable (FSMIE):** When set to 1, the SPI asserts an SMI# request whenever the Flash Cycle Done but is 1. |
| 30 | 0x0<br>RO | Reserved. |
| 29:24 | 0x0<br>RW | **Flash Data Byte Count (FDBC):** The field specifies the number of bytes to shift in or out during the data portion of the SPI cycle.<br>The contents of this register are 0s based with 0b representing 1 byte and 3Fh representing 64 bytes.<br>The number of bytes transferred is the value of this field plus 1.<br>The field is ignored for the Block Erase command. |
| 23:22 | 0x1<br>RO | Reserved. |
| 21 | 0x0<br>RW | **Write Enable Type (WET):** Write Enable Type (WET)<br>• 0 - Use 06h as the write enable instruction<br>• 1 - Use 50h as the write enable instruction |
| 20:17 | 0x0<br>RW | **FLASH Cycle (FCYCLE):** This field defines the Flash SPI cycle type generated to the FLASH when the FGO but is set as defined below:<br>• 0 - Read (1 up to 64 bytes by setting FDBC)<br>• 1 - Reserved<br>• 2 - Write (1 up to 64 bytes by setting FDBC)<br>• 3 - 4k Block Erase<br>• 4 - 64K Sector Erase<br>• 5 - Read SFDP<br>• 6 - Read JEDEC ID<br>• 7 - Write Status<br>• 8 - Read Status<br>• 9 - RPMC Op1<br>• A - RPMC Op2<br>Flash controller hardware automatically inserts a write enable opcode prior to write and erase operations.<br>Hardware automatically polls for device not-busy using read status after write and erase operations.<br>Flash controller hardware automatically polls for device not-ready using read status and RPMC Op2 after RPMC Op1 operations.<br>Read status is repeated until the device is not-busy, then RPMC Op2 is repeated until the device is not-busy.<br>If the device does not support 64k erase size (or if it doesn't support SFDP) then only 4k is allowed.<br>***Note:*** If reserved 1 is programmed to this field, flash controller will handle it as if it is 0 (Read). |
| 16 | 0x0<br>RW/1S/V | Flash Cycle Go (FGO): A write to this register with a 1 in this but initiates a request to the Flash SPI Arbiter to start a cycle.<br>This register is cleared by hardware when the cycle is granted by the SPU arbiter to run the cycle on the SPI bus. When the cycle is complete, the FDONE bit is set.<br>Software is forbidden to write to any register in the HSFLCTL register between the FGO bit getting set and the FDONE bit being cleared. Any attempt to violate this rule will be ignored by hardware.<br>Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1. This saves an additional memory write.<br>This but always returns 0 on reads. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW/L | **Flash Configuration Lock-Down (FLOCKDN):** When set to 1, those Flash Program Registers that are locked down by this FLOCKDN but cannot be written.<br>Once set to 1, this but can only be cleared by a hardware reset. |
| 14 | 0x0<br>RO/V | **Flash Descriptor Valid (FDV):** This bit is set to a 1 if the Flash Controller read the correct Flash Descriptor Signature.<br>If the Flash Descriptor Valid bit is not 1, software cannot use the Hardware Sequencing registers, but must use the software sequencing registers. Any attempt to use the Hardware Sequencing registers will result in the FCERR bit being set. |
| 13 | 0x1<br>RO/V | **Flash Descriptor Override Pin-Strap Status (FDOPSS):** This register reflects the value the Flash Descriptor Override Pin-Strap.<br>• 1 - No override<br>• 0 - The Flash Descriptor Override strap is set |
| 12 | 0x0<br>RW/L | **PRR3 PRR4 Lock-Down (PRR34_LOCKDN):** When set to 1, the BIOS PRR3 and PRR4 registers cannot be written.<br>Once set to 1, this bit can only be cleared by a hardware reset. |
| 11 | 0x0<br>RW/L | **Flash Configuration Lock-Down (WRSDIS):**<br>• 0 - Write status operation may be used using Hardware Sequencing<br>• 1 - Write status is not allowed as a Hardware Sequencing Operation. The flash controller will block the operation and set the FCERR but when software sets the “go” bit. |
| 10:6 | 0x0<br>RO | Reserved. |
| 5 | 0x0<br>RO/V | **SPI Cycle In Progress (H_SCIP):** hardware sets this bit when software sets the Flash Cycle Go (FGO) bit in the Hardware Sequencing Flash Control register.<br>This bit remains set until the cycle completes on the SPI interface. Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command.<br>Software must only program the next command when this bit is 0. |
| 4:0 | 0x0<br>RO | Reserved. |

## 60.3.3 Flash Address (BIOS_FADDR)—Offset 8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 8h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 26:0 | 0x0<br>RW | **Flash Linear Address (FLA):** The FLA is the starting byte linear address of a SPI Read or Write cycle or an address within a Block for the Block Erase command.<br>The Flash Linear Address must fall within a region for which BIOS has access permissions.<br>Hardware must convert the FLA into a Flash Physical Address (FPA) before running this cycle on the SPI bus. When operating in non-descriptor mode bit [26:24] is ignored and the FLA [23:0] is the FPA. |

## 60.3.4 Discrete Lock Bits (BIOS_DLOCK)—Offset Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RW | **Reserved (RSVD):** Whenever modifying BIOS_DLOCK read the value of bits 31:17 and write back the value read. |
| 16 | 0x0<br>RW/L | **SSEQ Lock-Down (SSEQLOCKDN):** IA FW Software Sequencing registers are locked when the logical OR of this bit and FLOCKDN is true.<br>The affected registers are BIOS_SSFSTS_CTL.SCF, BIOS_PREOP_OPTYPE, BIOS_OPMENU0, and BIOS_OPMENU1.<br>Once set to 1 this register is only cleared by host partition reset. |
| 15:13 | 0x0<br>RW | **Reserved (RSVD):** Whenever modifying BIOS_DLOCK read the value of bits 15:13 and write back the value read. |
| 12 | 0x0<br>RW/L | **PR4 Lock-Down (PR4LOCKDN):** IA FW PR4 register is locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 11 | 0x0<br>RW/L | **PR3 Lock-Down (PR3LOCKDN):** IA FW PR3 register is locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 10 | 0x0<br>RW/L | **PR2 Lock-Down (PR2LOCKDN):** IA FW PR2 register is locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 9 | 0x0<br>RW/L | **PR1 Lock-Down (PR1LOCKDN):** IA FW PR1 register is locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 8 | 0x0<br>RW/L | **PR0 Lock-Down (PR0LOCKDN):** IA FW PR0 register is locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 7:4 | 0x0<br>RW | **Reserved (RSVD):** Whenever modifying BIOS_DLOCK read the value of bits 7:4 and write back the value read. |
| 3 | 0x0<br>RW/L | **SBMRAG Lock-Down (SBMRAGLOCKDN):** IA FW SFRACC.BMRAG register bits are locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 2 | 0x0<br>RW/L | **SBMWAG Lock-Down (SBMWAGLOCKDN):** IA FW SFRACC.BMWAG register bits are locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 1 | 0x0<br>RW/L | **BMRAG Lock-Down (BMRAGLOCKDN):** IA FW FRACC.BMRAG register bits are locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |
| 0 | 0x0<br>RW/L | **BMWAG Lock-Down (BMWAGLOCKDN):** IA FW FRACC.BMWAG register bits are locked when the logical OR of this bit and FLOCKDN is true.<br>Once set to 1 this register is only cleared by host partition reset. |

## 60.3.5 Flash Data 0 (BIOS_FDATA0)—Offset 10h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 10h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 0 (FD0):** This field is shifted out as the SPI Data on the Master-Out Slave-In Data pin during the data portion of the SPI cycle.<br>This register also shifts in the data from the Master-In Slave-Out pin into this register during the data portion of the SPI cycle.<br>The data is always shifted starting with the least significant byte, msb to lsb, followed by the next least significant byte, msb to lsb, etc.<br>Specifically, the shift order on SPI in terms of bits within this register is: 7-6-5-4-3-2-1-0-15-14-13-?8-23-22-?16-31?24 Bit 24 is the last bit shifted out/in.<br>There are no alignment assumptions; byte 0 always represents the value specified by the cycle address.<br>***Note:*** The data in this register may be modified by the hardware during any programmed SPI transaction. Direct Memory Reads do not modify the contents of this register. |

## 60.3.6 Flash Data 1 (BIOS_FDATA1)—Offset 14h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 14h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 1 (FD1):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD0 has completely shifted in/out. |

## 60.3.7 Flash Data 2 (BIOS_FDATA2)—Offset 18h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 18h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 2 (FD2):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD1 has completely shifted in/out. |

## 60.3.8 Flash Data 3 (BIOS_FDATA3)—Offset 1Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 1Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 3 (FD3):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD2 has completely shifted in/out. |

## 60.3.9 Flash Data 4 (BIOS_FDATA4)—Offset 20h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 20h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 4 (FD4):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD3 has completely shifted in/out. |

## 60.3.10 Flash Data 5 (BIOS_FDATA5)—Offset 24h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 24h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 5 (FD5):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD4 has completely shifted in/out. |

## 60.3.11 Flash Data 6 (BIOS_FDATA6)—Offset 28h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 28h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 6 (FD6):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD5 has completely shifted in/out. |

## 60.3.12 Flash Data 7 (BIOS_FDATA7)—Offset 2Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 2Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 7 (FD7):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD6 has completely shifted in/out. |

## 60.3.13 Flash Data 8 (BIOS_FDATA8)—Offset 30h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 30h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 8 (FD8):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD7 has completely shifted in/out. |

## 60.3.14 Flash Data 9 (BIOS_FDATA9)—Offset 34h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 34h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 9 (FD9):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD8 has completely shifted in/out. |

## 60.3.15 Flash Data 10 (BIOS_FDATA10)—Offset 38h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 38h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 10 (FD10):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD9 has completely shifted in/out. |

## 60.3.16 Flash Data 11 (BIOS_FDATA11)—Offset 3Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 3Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 11 (FD11):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD10 has completely shifted in/out. |

## 60.3.17 Flash Data 12 (BIOS_FDATA12)—Offset 40h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 40h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 12 (FD12):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD11 has completely shifted in/out. |

## 60.3.18 Flash Data 13 (BIOS_FDATA13)—Offset 44h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 44h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 13 (FD13):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD12 has completely shifted in/out. |

## 60.3.19 Flash Data 14 (BIOS_FDATA14)—Offset 48h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 48h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 14 (FD14):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD13 has completely shifted in/out. |

## 60.3.20 Flash Data 15 (BIOS_FDATA15)—Offset 4Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 4Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Flash Data 15 (FD15):** Similar definition as Flash Data 0.<br>However, this register does not begin shifting until FD14 has completely shifted in/out. |

## 60.3.21 Flash Region Access Permissions (BIOS_FRACC)—Offset 50h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 000042C2h

**BAR and Offset:** [BIOS_SPI_BAR0] + 50h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/L | **BIOS Master Write Access Grant (BMWAG):** Each bit [31:24] corresponds to Master [7:0].<br>BIOS can grant one or more masters write access to the BIOS region 1 overriding the permissions in the Flash Descriptor.<br>The contents of this register are locked by the FLOCKDN bit. |
| 23:16 | 0x0<br>RW/L | **BIOS Master Read Access Grant (BMRAG):** Each bit [23:16] corresponds to Master [7:0].<br>BIOS can grant one or more masters read access to the BIOS region 1 overriding the read permissions in the Flash Descriptor.<br>The contents of this register are locked by the FLOCKDN bit. |
| 15:8 | 0x42<br>RO/V | **BIOS Region Write Access (BRWA):** Each bit [15:8] corresponds to Regions [7:0].<br>If the bit is set, this master can erase and write that particular region through register accesses.<br><br>The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Write Access OR a particular master has granted BIOS write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set.<br>BIOS always have the write access to its own Region 1 and Region 6 by default. Thus, the reset default of this field will be read as “42h”regardless of descriptor or non-descriptor mode. |
| 7:0 | 0xc2<br>RO/V | **BIOS Region Read Access (BRRA):** Each bit [7:0] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses.<br><br>The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Write Access OR a particular master has granted BIOS read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set.<br>BIOS always have the read access to its own Region 1 and Region 6 by default. Thus, the reset default of this field will be read as “42h” regardless of descriptor or non-descriptor mode. |

## 60.3.22 Flash Region 0 (BIOS_FREG0)—Offset 54h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 54h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 0 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 0 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Base. |

## 60.3.23 Flash Region 1 (BIOS_FREG1)—Offset 58h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 58h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 1 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 1 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Base. |

## 60.3.24 Flash Region 2 (BIOS_FREG2)—Offset 5Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 5Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 2 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 2 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Base. |

## 60.3.25 Flash Region 3 (BIOS_FREG3)—Offset 60h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 60h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 3 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 3 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Base. |

## 60.3.26 Flash Region 4 (BIOS_FREG4)—Offset 64h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 64h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 4 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 4 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Base. |

## 60.3.27 Flash Region 5 (BIOS_FREG5)—Offset 68h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 68h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 5 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Limit |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 5 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Base |

## 60.3.28 Flash Region 6 (BIOS_FREG6)—Offset 6Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 6Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 6 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Limit if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all zeros. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 6 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Base if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all ones. |

## 60.3.29 Flash Region 7 (BIOS_FREG7)—Offset 70h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 70h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 7 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 7 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Base. |

## 60.3.30 Flash Region 8 (BIOS_FREG8)—Offset 74h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 74h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 8 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 8 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Base. |

## 60.3.31 Flash Region 9 (BIOS_FREG9)—Offset 78h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00007FFFh

**BAR and Offset:** [BIOS_SPI_BAR0] + 78h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 9 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x7fff<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 9 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Base. |

## 60.3.32 Flash Region 10 (BIOS_FREG10)—Offset 7Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 7Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 10 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 10 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Base. |

## 60.3.33 Flash Region 11 (BIOS_FREG11)—Offset 80h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 80h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 11 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 11 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Base. |

## 60.3.34 Flash Protected Range 0 (BIOS_FPR0)—Offset 84h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 84h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RW/L | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RW/L | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RW/L | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.35 Flash Protected Range 1 (BIOS_FPR1)—Offset 88h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 88h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RW/L | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RW/L | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RW/L | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.36 Flash Protected Range 2 (BIOS_FPR2)—Offset 8Ch

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 8Ch

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RW/L | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RW/L | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RW/L | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.37 Flash Protected Range 3 (BIOS_FPR3)—Offset 90h

This register can not be written when the PRR34_LOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 90h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RW/L | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RW/L | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RW/L | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.38 Flash Protected Range 4 (BIOS_FPR4)—Offset 94h

This register can not be written when the PRR34_LOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 94h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/L | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RW/L | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RW/L | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RW/L | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.39 Global Protected Range 0 (BIOS_GPR0)—Offset 98h

This register is initialized via softstraps. This protected range applies globally to all masters / flash requesters.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 98h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.40 Software Sequencing Flash Status and Control (BIOS_SSFSTS_CTL)—Offset A0h

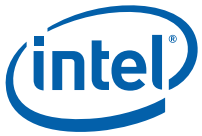
**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** FE000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + A0h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:27 | 0x1f<br>RO | **Reserved (RSVD):** This field is not programmable, it is fixed at 1Fh. |
| 26:24 | 0x6<br>RW/L | **SPI Cycle Frequency (SCF):**<br>The listed frequencies are approximate.<br>• 010: 17MHz<br>• 100: 30MHz<br>• 110: 48MHz<br>• All Others: Reserved<br>This register sets frequency to use for all SPI Software Sequencing cycles (write, erase, fast read, read status,.etc) except for the Read cycle which always run at 20MHz.<br>This register is locked when the SPI Configuration Lock-Down (FLOCKDN) bit is set. |
| 23 | 0x0<br>RW | **SPI SMI# Enable (SME):** When set to 1, the SPI asserts an SMI# request whenever the Cycle Done Status bit is 1. |
| 22 | 0x0<br>RW | **Data Cycle (DS):** When set to 1, there is data that corresponds to this transaction.<br>When 0, no data is delivered for this cycle, and the DBC and data fields themselves are don't cares. |
| 21:16 | 0x0<br>RW | **Data Byte Count (DBC):** This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The valid settings (in decimal) are any value from 0 to 63. The number of bytes transferred is the value of this field plus 1.<br>***Note:*** When this field is 00_0000b, then there is 1 byte to transfer and that 11_1111b means there are 64 bytes to transfer. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:12 | 0x0<br>RW | **Cycle Opcode Pointer (COP):** This field selects one of the programmed opcodes in the Opcode Menu to be used as the SPI Command/Opcode.<br>In the case of an Atomic Cycle Sequence, this determines the second command. |
| 11 | 0x0<br>RW | **Sequence Prefix Opcode Pointer (SPOP):** This field selects one of the two programmed prefix opcodes for use when performing an Atomic Cycle Sequence.<br>A value of 0 points to the opcode in the least significant byte of the Prefix Opcodes register.<br>By making this programmable, the ICH supports flash devices that have different opcodes for enabling writes to the data space vs. status register. |
| 10 | 0x0<br>RW | **Atomic Cycle Sequence (ACS):** When set to 1 along with the SCGO assertion, the SoC will execute a sequence of commands on the SPI interface without allowing another master to arbitrate and interleave cycles.<br>The sequence is composed of:<br>• Atomic Sequence Prefix Command (8-bit opcode only)<br>• Primary Command specified below by software (can include address and data)<br>• Polling the Flash Status Register (opcode 05h) until bit 0 becomes 0b.<br>The SPI Cycle in Progress bit remains set and the Cycle Done Status bit remains unset until the Busy bit in the Flash Status Register returns 0. |
| 9 | 0x0<br>RW/1S/V | **SPI Cycle Go (SCGO):** This bit always returns 0 on reads. However, a write to this register with a '1' in this bit starts the SPI cycle defined by the other bits of this register. The "SPI Cycle in Progress" (SCIP) bit gets set by this action.<br>Hardware must ignore writes to this bit while the Cycle In Progress bit is set.<br>Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1.<br>This saves an additional memory write. |
| 8 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |



| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO/V | **Fast Read Supported (FAST_READ_SUPPORT):** This bit reflects the value of the Fast Read Support bit in the Flash Descriptor Component Section. |
| 6 | 0x0<br>RO/V | **Dual Output Fast Read Supported (DUAL_FAST_READ_SUPPORT):** This bit reflects the value of the Dual Output Fast Read Support bit in the Flash Descriptor Component Section. |
| 5 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 4 | 0x0<br>RO/V | **Access Error Log (AEL):** This bit reflects the value of the Hardware Sequencing Status.AEL register. |
| 3 | 0x0<br>RW/1C/V | **Flash Cycle Error (FCERR):** Hardware sets this bit to 1 when a programmed access is blocked from running on the SPI interface due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress.<br>This bit remains asserted until cleared by software writing a 1 or hardware reset. |
| 2 | 0x0<br>RW/1C/V | **Cycle Done Status (CYCLE_DONE_STS):** The SoC sets this bit to 1 when the SPI Cycle completes (i.e., SCIP bit is 0) after software sets the GO bit. This bit remains asserted until cleared by software writing a 1 or hardware reset.<br>When this bit is set and the SPI SMI# Enable bit is set, an internal signal is asserted to the SMI# generation block.<br>Software must make sure this bit is cleared prior to enabling the SPI SMI# assertion for a new programmed access. |
| 1 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 0 | 0x0<br>RO/V | **SPI Cycle In Progress (SCIP):** Hardware sets this bit when software sets the SPI Cycle Go bit in the Command register. |

## 60.3.41 Prefix Opcode and Opcode Type Configuration (BIOS_PREOP_OPTYPE)—Offset A4h

This register is not writable when the Flash Configuration Lock-Down (FLOCKDN) bit is set. Entries in this register correspond to the entries in the Opcode Menu Configuration register.

Note that the definition below only provides write protection for opcodes that have addresses associated with them. Therefore, any erase or write opcodes that do not use an address should be avoided (for example, "Chip Erase" and "Auto-Address Increment Byte Program").

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + A4h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0 RW/L | **Opcode Type 7 (OPCODE_TYPE7):** See the description for bits 17:16. |
| 29:28 | 0x0 RW/L | **Opcode Type 6 (OPCODE_TYPE6):** See the description for bits 17:16. |
| 27:26 | 0x0 RW/L | **Opcode Type 5 (OPCODE_TYPE5):** See the description for bits 17:16. |
| 25:24 | 0x0 RW/L | **Opcode Type 4 (OPCODE_TYPE4):** See the description for bits 17:16. |
| 23:22 | 0x0 RW/L | **Opcode Type 3 (OPCODE_TYPE3):** See the description for bits 17:16. |
| 21:20 | 0x0 RW/L | **Opcode Type 2 (OPCODE_TYPE2):** See the description for bits 17:16. |
| 19:18 | 0x0 RW/L | **Opcode Type 1 (OPCODE_TYPE1):** See the description for bits 17:16. |
| 17:16 | 0x0 RW/L | **Opcode Type 0 (OPCODE_TYPE0):** This field specifies information about the corresponding Opcode 0. This information allows the hardware to 1) know whether to use the address field and 2) provide BIOS and Shared Flash protection capabilities.The hardware implementation also uses the read vs. write information for modifying the behavior of the SPI interface logic. The encoding of the two bits is:<br>• 00 = No Address associated with this Opcode and Read Cycle type<br>• 01 = No Address associated with this Opcode and Write Cycle type<br>• 10 = Address required; Read cycle type<br>• 11 = Address required; Write cycle type |
| 15:8 | 0x0 RW/L | **Prefix Opcode 1 (PREFIX_OPCODE1):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence. |
| 7:0 | 0x0 RW/L | **Prefix Opcode 0 (PREFIX_OPCODE0):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence. |

## 60.3.42 Opcode Menu0 Configuration (BIOS_OPMENU0)—Offset A8h

This register is not writable when the SPI Configuration Lock-Down bit (FLOCKDN) is set. Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices.

It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + A8h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/L | **Allowable Opcode 3 (OPCODE3):** See the description for bits 7:0. |
| 23:16 | 0x0<br>RW/L | **Allowable Opcode 2 (OPCODE2):** See the description for bits 7:0. |
| 15:8 | 0x0<br>RW/L | **Allowable Opcode 1 (OPCODE1):** See the description for bits 7:0. |
| 7:0 | 0x0<br>RW/L | **Allowable Opcode 0 (OPCODE0):** Software programs an SPI opcode into this field for use when initiating SPI commands through the Control Register. |

## 60.3.43 Opcode Menu1 Configuration (BIOS_OPMENU1)—Offset ACh

This register is not writable when the SPI Configuration Lock-Down bit (FLOCKDN) is set. Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices.

It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + ACh

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/L | **Allowable Opcode 7 (OPCODE7):** See the description for bits 7:0. |
| 23:16 | 0x0<br>RW/L | **Allowable Opcode 6 (OPCODE6):** See the description for bits 7:0. |
| 15:8 | 0x0<br>RW/L | **Allowable Opcode 5 (OPCODE5):** See the description for bits 7:0. |
| 7:0 | 0x0<br>RW/L | **Allowable Opcode 4 (OPCODE4):** Software programs an SPI opcode into this field for use when initiating SPI commands through the Control Register. |

## 60.3.44 Secondary Flash Region Access Permissions (BIOS_SFRACC)—Offset B0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + B0h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RW/L | **Secondary BIOS Master Write Access Grant (SECONDARYBIOS_MWAG):** Each bit [31:29] corresponds to Master [7:0]. BIOS can grant one or more masters write access to the Secondary BIOS region 6 overriding the permissions in the Flash Descriptor. The contents of this register are locked by the FLOCKDN bit. |
| 23:16 | 0x0<br>RW/L | **Secondary BIOS Master Read Access Grant (SECONDARYBIOS_MRAG):** Each bit [28:16] corresponds to Master [7:0]. BIOS can grant one or more masters read access to the Secondary BIOS region 6 overriding the read permissions in the Flash Descriptor. The contents of this register are locked by the FLOCKDN bit. |
| 15:0 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |

## 60.3.45 Flash Descriptor Observability Control (BIOS_FDOC)—Offset B4h

This is a test mode only register that can be used to observe the contents of the Flash Descriptor that is stored internally in the SoC Flash Controller. The CPU Complex soft straps are not observable in SoC as these are forwarded to the CPU Complex and not stored.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + B4h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:15 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 14:12 | 0x0<br>RW | **Flash Descriptor Section Select (FDSS):** Selects which section within the loaded Flash Descriptor to observe.<br>• 000: Flash Signature and Descriptor Map<br>• 001: Component<br>• 010: Region<br>• 011: Master<br>• 100: Soft Straps<br>• Other: Reserved |
| 11:2 | 0x0<br>RW | **Flash Descriptor Section Index (FDSI):** Selects the DW offset within the Flash Descriptor Section to observe. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |

## 60.3.46 Flash Descriptor Observability Data (BIOS_FDOD)—Offset B8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + B8h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Flash Descriptor Section Data (FDSD):** Returns the DW of data to observe as selected in the Flash Descriptor Observability Control. |

## 60.3.47 Additional Flash Control (BIOS_AFC)—Offset C0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + C0h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 9:1 | 0x0<br>RW | **Reserved RW Scratch Pad (Reserved_SCRATCHPAD):** Scratch Pad bits that are R/W to be used |
| 0 | 0x0<br>RW/V/P | **Stop Prefetch on Flush Pending (SPFP):** When set to '1', the in progress of a prefetch will be ended if subsequence access from the master of the same interface (SPXB or MBB) is detected to be a cache-miss and read cache will be flushed. When set to '0', the prefetch will be allowed to complete prior to flushing.<br>This register is only reset on a Early boot reset of both the Host and the ME partitions.NOTE: This should change to reset on a host reset and a different SPFP bit used for the Intel® Management Engine cache. Future<br>project fix?<br>The reset value of this bit is determined by the SoC soft strap PCHSTRP0.SPI_SPFP. |

## 60.3.48 Vendor Specific Component Capabilities for Component 0 (BIOS_SFDP0_VSCC0)—Offset C4h

*Note:* Intel is planning to deprecate support for non-SFDP capable flash devices.

VSCC registers were previously used to allow software to configure flash device-specific parameters. Since Serial Flash Discoverable Parameters were standardized in 2011 this capability is less important. For SPT, the VSCC registers will be renamed SFDP registers. If the flash controller discovers SFDP devices, then hardware will populate the appropriate register fields from the SFDP table and treat them as read-only by software. If SFDP is not discovered then the registers will remain writable by software. This change is intended to allow microarchitectural cleanup by having device parameters sourced from a single location. The SPI flash controller uses the Quad Enable Requirements (QER) data from the SFDP table to ensure that the flash device is enabled for Quad-mode prior to issuing any Quad-mode reads. If quad reads are disabled via softstraps then the flash controller may ignore QER. If the QE bit is in the flash status register, the controller reads the status register. If the QE bit is 0 then the controller sets it to 1 and writes the status register back to the flash device using a write_enable, write_status sequence.

The QER bits are used by software/firmware as follows:

1. Manufacturing time: FITC tool is used to initialize the set of vendor-specific VSCC entries in the FLUMAP portion of the descriptor, where each entry is a pair of (device ID, VSCC value).
2. Boot time: Intel® Management Engine firmware and BIOS perform the SoC register initialization.
   a. Read JEDEC ID from each flash device.
   b. Scan through VSCC tables pointed to by FLUMAP.
   c. Copy the entry from FLUMAP which matches device 0 into VSCC0, copy entry which matches device 1 into VSCC1.
3. In software sequencing, BIOS/Intel® Management Engine ensure QE bit is not cleared while doing status register write.

The flash controller uses the polling delays to set internal timers after initiating a write, erase, or RPMC operation. After the internal timer expires the flash controller begins polling the flash busy status in bit 0 of the status register. [OPEN: Move requirement to a standalone section] If the RPMC polling requirements bit is set to poll with RPMC Op2, then the flash controller issues an RPMC Op2 instead of a read status (05h). The RPMC Op2 operation uses “fast read” timing with 8 dummy bits between the address and valid data. Bit 0 of the first byte returned from RPMC Op2 is the busy bit.

The fields in this register pertain to cycles targeting addresses less than FLCOMP.C0DEN, i.e. within Component 0.

If the CPPTV bit is '0', software must configure the VSCC register appropriately. If the CPPTV bit is '1', the corresponding parameter values discovered via SFDP will be used. In most cases, software is not required to configure the VSCC register. However, if the SFDP table indicates an erase size other than 4k Byte, then software is required to program the VSCC.EO register with the correct erase opcode.

If the CPPTV bit is '0' and the VSCC register is not programmed and the Flash is operating in Flash Mode (as opposed to EEPROM mode), then software cannot use hardware sequencing. This requirement is not enforced by hardware, but is a requirement of the power-on usage model.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00002000h

**BAR and Offset:** [BIOS_SPI_BAR0] + C4h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 0.<br>***Note:*** If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery. |
| 30 | 0x0<br>RW/L | **Vendor Component Lock (VCL):**<br>0 = The lock bit is not set.<br>1 = The Vendor Component Lock bit is set. This register locks itself when set. |
| 29 | 0x0<br>RW/V/L | **64k Erase Valid (EO_64k_VALID):**<br>0 = The EO_64k opcode is not valid.<br>1 = The EO_64k opcode is valid. |
| 28 | 0x0<br>RW/V/L | **4k Erase Valid (EO_4k_VALID):**<br>0 = The EO_4k opcode is not valid.<br>1 = The EO_4k opcode is valid. |
| 27 | 0x0<br>RW/L | **RPMC Supported (RPMC_SUPPORTED):**<br>0 = The device does not support RPMC.<br>1 = The device supports RPMC. |
| 26 | 0x0<br>RW/V/L | **Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED):**<br>0 = The device does not support Deep Powerdown.<br>1 = The device supports Deep Powerdown. |
| 25 | 0x0<br>RW/V/L | **Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED):**<br>0 = The device does not support Suspend/Resume.<br>1 = The device supports Suspend/Resume. |
| 24 | 0x0<br>RW/V/L | **Soft Reset Supported (SOFT_RST_SUPPORTED):**<br>0 = The device does not support Soft Reset.<br>1 = The device supports Soft Reset. |
| 23:16 | 0x0<br>RW/V/L | **64k Erase Opcode (EO_64k):**<br>This register is programmed with the Flash 64k sector erase instruction opcode for component 0.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. |
| 15:8 | 0x20<br>RW/V/L | **4k Erase Opcode (EO_4k):** This register is programmed with the Flash 4k subsector erase instruction opcode for component 0. Software must program this register if the SFDP table for this component does not show 4 k Byte erase capability.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. |
| 7:5 | 0x0<br>RW/V/L | **Quad Enable Requirements (QER):**<br>• 000 Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability.<br>• 001 Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes.<br>• 010 Part requires bit 6 of status register 1 to be set to enable quad IO.<br>• 011 Part requires bit 7 of the configuration register to be set to enable Quad.<br>• 100 Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte.<br>This register is locked by the Vendor Component Lock (VCL) bit.<br>If the SFDP table contains this information, the flash controller loads these bits from the table.<br>The flash controller uses this information to prevent clearing the QE bit in the flash device's status register when WSR=1. |
| 4 | 0x0<br>RW/V/L | **Write Enable on Write Status (WEWS):**<br>0 = 50h is the opcode to enable a status register write.<br>1 = 06h is the opcode to enable a status register write.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/V/L | **Write Status Required (WSR):**<br>0 = No requirement to write to the Write Status Register prior to a write<br>1 = A write is required to the Write Status Register prior to write and erase to remove any protection. This is required for SST components.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |
| 2 | 0x0<br>RW/V/L | **Write Granularity (WG):**<br>0 = Reserved.<br>1 = 64 Byte.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |
| 1:0 | 0x0<br>RW/L | **Reserved (RSVD):** Reserved. |

## 60.3.49 Vendor Specific Component Capabilities for Component 1 (BIOS_SFDP1_VSCC1)—Offset C8h

*Note:* Intel is planning to deprecates support for non-SFDP capable flash devices.

See 31.1, Section 4)a). The fields in this register pertain to cycles targeting addresses greater than or equal to FLCOMP.C0DEN, i.e. outside of Component 0. The lockable bits in this register are locked when either CPPTV is set to '1' by hardware or when VCL is '1'.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + C8h

**Default:** 00002000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 1.<br>***Note:*** If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery. |
| 30 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 29 | 0x0<br>RW/V/L | **64k Erase Valid (EO_64k_VALID):**<br>0 = The EO_64k opcode is not valid.<br>1 = The EO_64k opcode is valid. |
| 28 | 0x0<br>RW/V/L | **4k Erase Valid (EO_4k_VALID):**<br>0 = The EO_4k opcode is not valid.<br>1 = The EO_4k opcode is valid. |
| 27 | 0x0<br>RW/L | **RPMC Supported (RPMC_SUPPORTED):**<br>0 = The device does not support RPMC.<br>1 = The device supports RPMC. |
| 26 | 0x0<br>RW/V/L | **Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED):**<br>0 = The device does not support Deep Powerdown.<br>1 = The device supports Deep Powerdown. |
| 25 | 0x0<br>RW/V/L | **Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED):**<br>0 = The device does not support Suspend/Resume.<br>1 = The device supports Suspend/Resume. |
| 24 | 0x0<br>RW/V/L | **Soft Reset Supported (SOFT_RST_SUPPORTED):**<br>0 = The device does not support Soft Reset.<br>1 = The device supports Soft Reset. |
| 23:16 | 0x0<br>RW/V/L | **64k Erase Opcode (EO_64k):** This register is programmed with the Flash 64k sector erase instruction opcode for component 1.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. |
| 15:8 | 0x20<br>RW/V/L | **4k Erase Opcode (EO_4k):** This register is programmed with the Flash 4k subsector erase instruction opcode for component 1.<br>Software must program this register if the SFDP table for this component does not show 4 k Byte erase capability.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>RW/V/L | **Quad Enable Requirements (QER):**<br>• 000 Part does not require a Quad Enable bit to be set, either because Quad is not supported or because the manufacturer somehow permanently enables Quad capability.<br>• 001 Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to status register clears all bits in register 2, therefore status register writes MUST be two bytes.<br>• 010 Part requires bit 6 of status register 1 to be set to enable quad IO.<br>• 011 Part requires bit 7 of the configuration register to be set to enable Quad.<br>• 100 Part requires bit 9 in status register 2 to be set to enable quad IO. Writing one byte to the status register does not clear the second byte.<br>This register is locked by the Vendor Component Lock (VCL) bit.<br>If the SFDP table contains this information, the flash controller loads these bits from the table.<br>The flash controller uses this information to prevent clearing the QE bit in the flash device's status register when WSR=1. |
| 4 | 0x0<br>RW/V/L | **Write Enable on Write Status (WEWS):**<br>0 = 50h is the opcode to enable a status register write.<br>1 = 06h is the opcode to enable a status register write.<br>Must be set to '1' for Intel's Blanshard Flash Component and for Atmel. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |
| 3 | 0x0<br>RW/V/L | **Write Status Required (WSR):**<br>0 = No requirement to write to the Write Status Register prior to a write.<br>1 = A write is required to the Write Status Register prior to write and erase to remove any protection. This is required for SST components.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |
| 2 | 0x0<br>RW/V/L | **Write Granularity (WG):**<br>0 = Reserved.<br>1 = 64 Byte.<br>This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.<br>***Note:*** Hardware ignores the state of this bit. |
| 1:0 | 0x0<br>RW/L | **Reserved (RSVD_1):** Reserved. |

## 60.3.50 Parameter Table Index (BIOS_PTINX)—Offset CCh

Observability control for Component Property Tables.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + CCh

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15:14 | 0x0<br>RW | **Supported Parameter Table (SPT):** Selects which supported parameter table to observe.<br>• 00: Component 0 Property Parameter Table.<br>• 01: Component 1 Property Parameter Table.<br>• 10: Reserved.<br>• 11: Reserved. |
| 13:12 | 0x0<br>RW | **Header or Data (HORD):** Select parameter table header DW vs Data DW.<br>• 00: SFDP Header.<br>• 01: Parameter Table Header.<br>• 10: Data.<br>• 11: Reserved. |
| 11:2 | 0x0<br>RW | **Parameter Table DW Index (PTDWI):** Selects the DW offset within the parameter table to observe. |
| 1:0 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |

## 60.3.51 Parameter Table Data (BIOS_PTDATA)—Offset D0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + D0h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RO/V | **Parameter Table DW Data (PTDWD):** Returns the DW of data to observe as selected in the Parameter Table Index register.<br>***Note:*** The returned value of reserved fields in the SFDP header or table must be ignored by software.<br>The flash controller may return either 0 or 1 for these fields. |

## 60.3.52 SPI Bus Requester Status (BIOS_SBRS)—Offset D4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + D4h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **TPM access ongoing (TPM_ACC_ONG):** |
| 30:18 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 17:15 | 0x0<br>RO/V | **Master 6 Status (M6STATUS):** See description under M1STATUS. |
| 14:12 | 0x0<br>RO/V | **Master 5 Status (M5STATUS):** See description under M1STATUS. |
| 11:9 | 0x0<br>RO/V | **Master 4 Status (M4STATUS):** See description under M1STATUS. |
| 8:6 | 0x0<br>RO/V | **Master 3 Status (M3STATUS):** See description under M1STATUS. |
| 5:3 | 0x0<br>RO/V | **Master 2 Status (M2STATUS):** See description under M1STATUS. |
| 2:0 | 0x0<br>RO/V | **Master 1 Status (M1STATUS):** Indicates whether this master has an outstanding transaction enqueued or in flight and the transaction type.<br>• 0xx: No transaction.<br>• 100: Flash read transaction.<br>• 101: Flash write transaction.<br>• 110: Flash erase transaction.<br>• 111: Flash RPMC transaction. |

## 60.3.53 Flash Region 12 (BIOS_FREG12)—Offset E0h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + E0h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 12 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 12 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Base. |

## 60.3.54 Flash Region 13 (BIOS_FREG13)—Offset E4h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + E4h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 13 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 13 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Base. |

## 60.3.55 Flash Region 14 (BIOS_FREG14)—Offset E8h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + E8h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 14 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 14 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Base. |

## 60.3.56 Flash Region 15 (BIOS_FREG15)—Offset ECh

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + ECh

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 30:16 | 0x0<br>RO/V | **Region Limit (RL):** This specifies address bits 26:12 for the Region 15 Limit.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Limit. |
| 15 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 14:0 | 0x0<br>RO/V | **Region Base (RB):** This specifies address bits 26:12 for the Region 15 Base.<br>The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Base. |

## 60.3.57 Set Strap Msg Lock (SSML)—Offset F0h

Lockable: Yes

Power Well: Resume

reset_type= wrsmrst#.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + F0h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 0 | 0x0<br>RW/L | **Set_Strap Lock (SSL):** When set to '1', all of SSML, SSMC and SSMD is locked, including this Lock bit.<br>Note that this bit is reset on CF9 resets.<br>reset_type= wrsmrst# |

## 60.3.58 Set Strap Msg Control (SSMC)—Offset F4h

Lockable: Yes

Power Well: DSW

reset_type= dsw_pok

The DSW bits are all cleared by dsw_pok, but must not be cleared by CF9h resets.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + F4h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 0 | 0x0<br>RW/L | **Set_Strap Mux Select (SSMS):** When set to '1', the Set strap message bits [47:32] come from the Set_Strap Msg Data register.<br>This bit is reset by the RSMRST# pin only. When '0', the Set-Strap data continues to come from the soft straps themselves.<br>reset_type=dsw_pok<br>This register field is locked by the Set Strap Lock (SSML.SSL) bit. |

## 60.3.59 Set Strap Msg Data (SSMD)—Offset F8h

Lockable: Yes
Power Well: DSW
Reset Type: dsw_pok

This register is used to provide a BIOS programmable sticky register which contains data that will be used in the Set-Strap type 1 msg on subsequent resets. The bits in the message control certain CPU features, for which see the CPU spec. These bits are in the resume well, so only reset on G3.

The usage model is that on each reset BIOS will check the state of the CPU. If the state is correct, then BIOS continues. If not, then BIOS writes the SSMD and SSMC registers and does a CF9 reset. On the reset the value of what was written to SSMD takes effect. Note that some mobile platforms force G3 on S5 requests. For those platforms, if the user/BIOS wants to have these bits set, there will be 2 resets on every power-on. If the platform accepts the default of '0' for these controls, then there is only one reset. The bits are in DSW and not RTC well because this allows a user upgrade, assuming the user unplugged the system before doing the upgrade, to revert to a setting of 0. This should reduce any interoperability concerns regarding user upgrades. The DSW bits are all cleared dsw_pok, and must not be cleared by CF9h resets.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + F8h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15:0 | 0x0<br>RW/L | **Set_Strap Data (SSD):** When SSMS is '1', then this data is sent in the Set-Strap msg Type 1 upon reset. This data is sent in the 2nd DW of data, bits[15:0].<br>reset_type=int_sus_pok<br>This register field is locked by the Set Strap Lock (SSML.SSL) bit. |

## 60.3.60 RPMC SFDP Table (BIOS_RPMC0_D0)—Offset 108h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 108h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RPMC SFDP Table dword 0 Device 0 (RPMC0_D0):** |

## 60.3.61 RPMC SFDP Table (BIOS_RPMC1_D0)—Offset 10Ch

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 10Ch

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RPMC SFDP Table dword 1 Device 0 (RPMC1_D0):** |

## 60.3.62 RPMC SFDP Table (BIOS_RPMC0_D1)—Offset 110h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 110h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RPMC SFDP Table dword 0 Device 1 (RPMC0_D1):** |

## 60.3.63 RPMC SFDP Table (BIOS_RPMC1_D1)—Offset 114h

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**Default:** 00000000h

**BAR and Offset:** [BIOS_SPI_BAR0] + 114h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **RPMC SFDP Table dword 0 Device 1 (RPMC1_D1):** |

## 60.3.64 BIOS Master Read Access Permissions (BIOS_BM_RAP)—Offset 118h

This register is implemented in DNV, LBG and subsequent programs.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 118h

**Default:** 000000C2h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15:0 | 0xc2<br>RO/V | **BIOS Master Read Access Permissions (BMRAP) (BMRAP):** Each bit [15:0] corresponds to Regions [15:0].<br>If the bit is set, this master can read that particular region through register accesses.<br>The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Read Access OR a particular master has granted BIOS read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set.<br>BIOS has read access to its own Region 1, 6 and 7 by default. Thus, the reset default of this field will be read as "C2h". |

## 60.3.65 BIOS Master Write Access Permissions (BIOS_BM_WAP)—Offset 11Ch

This register is implemented in DNV, LBG and subsequent programs.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 11Ch

**Default:** 00000042h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD):** Reserved. |
| 15:0 | 0x42<br>RO/V | **BIOS Master Write Access Permissions (BMWAP) (BMWAP):** Each bit [15:0] corresponds to Regions [15:0]. If the bit is set, this master can erase and write that particular region through register accesses.<br>The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Write Access OR a particular master has granted BIOS write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set.<br>BIOS has write access to its own Region 1 and 6 by default. Thus, the reset default of this field will be read as "42h". |

## 60.3.66 IE Flash Protected Range 0 (BIOS_IE_PR0)—Offset 1B0h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 1B0h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.67 IE Flash Protected Range 1 (BIOS_IE_PR1)—Offset 1B4h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 1B4h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.68 IE Flash Protected Range 2 (BIOS_IE_PR2)—Offset 1B8h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 1B8h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison.<br>Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.69 IE Flash Protected Range 3 (BIOS_IE_PR3)—Offset 1BCh

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 1BCh

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range.<br>Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range.<br>Address bits 11:0 are assumed to be 000h for the base comparison.<br>Any address less than the value programmed in this field is unaffected by this protected range. |

## 60.3.70 IE Flash Protected Range 4 (BIOS_IE_PR4)—Offset 1C0h

This register can not be written when the FLOCKDN bit is set to 1.

**Type:** Memory Mapped I/O Register (Size: 32 bits)

**BAR and Offset:** [BIOS_SPI_BAR0] + 1C0h

**Default:** 00000000h

**BIOS_SPI_BAR0 Type:** PCI Configuration Register (Size: 32 bits)
**BIOS_SPI_BAR0 Reference:** [B:0, D:31, F:5] + 10h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO/V | **Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware.<br>The base and limit fields are ignored when this bit is cleared. |
| 30:16 | 0x0<br>RO/V | **Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range. |
| 15 | 0x0<br>RO/V | **Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that read directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared. |
| 14:0 | 0x0<br>RO/V | **Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range. |

§ §

# 61 Programmable Interval Timer - Legacy

## 61.1 Introduction and Index

The host-accessible registers for the Programmable Interval Timer (PIT) are described here.

The legend for the register access codes are described in Chapter 43, “Introduction.”

### 61.1.1 Host I/O Space—Fixed Address

An index of the registers that are accessible in Host I/O Space is shown here.

## 61.2 Registers in I/O Space—Fixed Addresses

### Table 61-1. Summary of I/O Registers—Fixed I/O Ports

| I/O Port (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 40 | 1 | C4 | "Counter 0 - Interval Timer Status Byte Format Register (C0_ITSBFR)—Offset 40h" on page 2876 |
| 42 | 1 | C4 | "Counter 2 - Interval Timer Status Byte Format Register (C2_ITSBFR)—Offset 42h" on page 2877 |
| 43 | 1 | 00 | "Timer Control Word Register (TCW)—Offset 43h" on page 2878 |
| 42 | 1 | 00 | "Counter 2 - Counter Access Ports Register (C2_CAPR)—Offset 42h" on page 2879 |
| 43 | 1 | C0 | "Read Back Command (RBC)—Offset 43h" on page 2880 |
| 43 | 1 | 00 | "Counter Latch Command (CLC)—Offset 43h" on page 2881 |
| 61 | 1 | 00 | "NMI Status and Control (NMI_STS_CNT)—Offset 61h" on page 2882 |

These registers are accessed in Host I/O Space through fixed I/O Port addresses.

## 61.2.1 Counter 0 - Interval Timer Status Byte Format Register (C0_ITSBFR)—Offset 40h

Each counter's status byte can be read following a Read Back Command. If latch status is chosen (bit 4=0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register (I/O Port 40h for counter 0 and I/O Port 42h for counter 2) returns the status byte. The status byte returns the following:

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 40h

**Default:** C4h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Counter OUT Pin State (COPS):**<br>• When this bit is a 1, the OUT pin of the counter is also a 1.<br>• When this bit is a 0, the OUT pin of the counter is also a 0.<br><br>**Power Well:** Core |
| 6 | 0x1<br>RO | **Count Register Status (CRSTS):** This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE).<br>The exact time this happens depends on the counter mode, but until the count is loaded into the counting element (CE), the count value will be incorrect.<br>• 0 - Count has been transferred from CR to CE and is available for reading.<br>• 1 - Null Count. Count has not been transferred from CR to CE and is not yet available for reading.<br><br>**Power Well:** Core |
| 5:4 | 0x0<br>RO | **Read/Write Selection Status (RW_SLT_STS):** These reflect the read/write selection made through bits[5:4] of the control register.<br>The binary codes returned during the status read match the codes used to program the counter read/write selection.<br>• 00 - Counter Latch Command<br>• 01 - Read/Write Least Significant Byte (LSB)<br>• 10 - Read/Write Most Significant Byte (MSB)<br>• 11 - Read/Write LSB then MSB<br><br>**Power Well:** Core |
| 3:1 | 0x2<br>RO | **Mode Selection Status (MD_SLT_STS):** These bits return the counter mode programming.<br>The binary code returned matches the code used to program the counter mode, as listed under the bit function above.<br>• 000 - Out signal on end of count (=0)<br>• 001 - Hardware retriggerable one-shot<br>• x10 - Rate generator (divide by n counter)<br>• x11 - Square wave output<br>• 100 - Software triggered strobe<br>• 101 - Hardware triggered strobe<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Countdown Type Status (CDT_STS):** This bit reflects the current countdown type, ether 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.<br><br>**Power Well:** Core |

## 61.2.2 Counter 2 - Interval Timer Status Byte Format Register (C2_ITSBFR)—Offset 42h

Each counter's status byte can be read following a Read Back Command. If latch status is chosen (bit 4=0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register (I/O Port 40h for counter 0 and I/O Port 42h for counter 2) returns the status byte. The status byte returns the following:

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 42h

**Default:** C4h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RO | **Counter OUT Pin State (COPS):**<br>• When this bit is a 1, the OUT pin of the counter is also a 1.<br>• When this bit is a 0, the OUT pin of the counter is also a 0.<br><br>**Power Well:** Core |
| 6 | 0x1<br>RO | **Count Register Status (CRSTS):** This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE).<br>The exact time this happens depends on the counter mode, but until the count is loaded into the counting element (CE), the count value will be incorrect.<br>• 0 - Count has been transferred from CR to CE and is available for reading.<br>• 1 - Null Count. Count has not been transferred from CR to CE and is not yet available for reading.<br><br>**Power Well:** Core |
| 5:4 | 0x0<br>RO | **Read/Write Selection Status (RW_SLT_STS):** These reflect the read/write selection made through bits[5:4] of the control register.<br>The binary codes returned during the status read match the codes used to program the counter read/write selection.<br>• 00 - Counter Latch Command<br>• 01 - Read/Write Least Significant Byte (LSB)<br>• 10 - Read/Write Most Significant Byte (MSB)<br>• 11 - Read/Write LSB then MSB<br><br>**Power Well:** Core |
| 3:1 | 0x2<br>RO | **Mode Selection Status (MD_SLT_STS):** These bits return the counter mode programming. The binary code returned matches the code used to program the counter mode, as listed under the bit function above.<br>• 000 - Out signal on end of count (=0)<br>• 001 - Hardware retriggerable one-shot<br>• x10 - Rate generator (divide by n counter)<br>• x11 - Square wave output<br>• 100 - Software triggered strobe<br>• 101 - Hardware triggered strobe<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Countdown Type Status (CDT_STS):** This bit reflects the current countdown type, ether 0 for binary countdown or a 1 for binary coded decimal (BCD) countdown.<br><br>**Power Well:** Core |

### 61.2.3 Timer Control Word Register (TCW)—Offset 43h

This register is programmed prior to any counter being accessed to specify counter modes. Following reset, the control words for each register are undefined and each counter output is 0. Each timer must be programmed to bring it into a known state. There are two special commands that can be issued to the counters through this register, the Read Back Command and the Counter Latch Command. When these commands are chosen, several bits within this register are redefined.

**Type:** I/O Register (Size: 8 bits)

**I/O Port:** 43h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Counter Select (CNT_SLT):** The Counter Selection bits select the counter the control word acts upon as shown below.<br>The Read Back Command is selected when bits[7:6] are both 1.<br>• 00 - Counter 0 select<br>• 01 - Reserved<br>• 10 - Counter 2 select<br>• 11 - Read Back Command<br><br>**Power Well:** Core |
| 5:4 | 0x0<br>WO | **Read/Write Select: (RW_SLT):** These bits are the read/write control bits.<br>The actual counter programming is done through the counter port (40h for counter 0 and 42h for counter 2).<br>• 00 - Counter Latch Command<br>• 01 - Read/Write Least Significant Byte (LSB)<br>• 10 - Read/Write Most Significant Byte (MSB)<br>• 11 - Read/Write LSB then MSB<br><br>**Power Well:** Core |
| 3:1 | 0x0<br>WO | **Counter Mode Selection (CNT_MD_SLTN):** These bits select one of six possible modes of operation for the selected counter.<br>• 000 - Out signal on end of count (=0)<br>• 001 - Hardware retriggerable one-shot<br>• x10 - Rate generator (divide by n counter)<br>• x11 - Square wave output<br>• 100 - Software triggered strobe<br>• 101 - Hardware triggered strobe<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Binary/BCD Countdown Select (B_BCD_CNTDWN_SLT):**<br>• 0 - Binary countdown is used. The largest possible binary count is $2^{16}$<br>• 1 - Binary coded decimal (BCD) count is used.The largest possible BCD count is $10^4$<br><br>**Power Well:** Core |

## 61.2.4 Counter 2 - Counter Access Ports Register (C2_CAPR)—Offset 42h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 42h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW/V | **Counter Port (CP):** Each counter port address is used to program the 16-bit Count Register.<br>The order of programming, either LSB only, MSB only, or LSB then MSB, is defined with the Interval Counter Control Register at I/O Port 43h.<br>The counter port is also used to read the current count from the Count Register, and return the status of the counter programming following a Read Back Command.<br><br>**Power Well:** Core |

## 61.2.5 Read Back Command (RBC)—Offset 43h

The Read Back Command is used to determine the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters.

**Type:** I/O Register (Size: 8 bits)

**I/O Port:** 43h

**Default:** C0h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x3<br>WO | **Read Back Command (RBC):** Must be 11 to select the Read Back Command (RBC). Otherwise, see "Counter Latch Command (CLC)—Offset 43h" on page 2881.<br><br>**Power Well:** Core |
| 5 | 0x0<br>WO | **Latch Count of Selected Counters (LCSC):**<br>• 0 - Current count value of the selected counters will be latched<br>• 1 - Current count will not be latched<br><br>**Power Well:** Core |
| 4 | 0x0<br>WO | **Latch Status of Selected Counters (LSSC):**<br>• 0 - Status of the selected counters will be latched<br>• 1 - Status will not be latched<br><br>**Power Well:** Core |
| 3 | 0x0<br>WO | **Counter 2 Select (CNT_2_SLT):**<br>When set to 1, Counter 2 count and/or status will be latched<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 1 | 0x0<br>WO | **Counter 0 Select (CNT_0_SLT):**<br>When set to 1, Counter 0 count and/or status will be latched.<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Reserved (RSVD_1):** Must be 0.<br><br>**Power Well:** Core |

### 61.2.6 Counter Latch Command (CLC)—Offset 43h

The Counter Latch Command latches the current count value. This command is used to insure that the count read from the counter is accurate.

**Type:** I/O Register (Size: 8 bits)

**I/O Port:** 43h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>WO | **Counter Selection (CNT_SLT):** These bits select the counter for latching.<br>• 00 - Counter 0<br>• 01 - Reserved<br>• 10 - Counter 2<br>***Note:*** If 11 is written, then the write is interpreted as a Read Back Command (RBC). See "Read Back Command (RBC)—Offset 43h" on page 2880.<br><br>**Power Well:** Core |
| 5:4 | 0x0<br>WO | **Counter Latch Command (CLC):** Write 00 to select the Counter Latch Command.<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>WO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |

## 61.2.7 NMI Status and Control (NMI_STS_CNT)—Offset 61h

NMI Status and Control Register

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 61h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **SERR# NMI Source Status (SERR_NMI_STS):** This bit is set by any of the sources of the internal SERR signal of the SoC. This includes SERR assertions forwarded from the secondary PCI bus, error from a PCIe port, or internal Bus 0 functions that generate SERR#.<br>Bit 2 must be cleared in this register in order for this bit to be set. This interrupt source is enabled by setting bit 2 to 0.<br>To reset the interrupt, set bit 2 to 1 and then set it to 0.<br>This bit is read-only. When writing to port 61h, this bit must be 0.<br><br>**Power Well:** Core |
| 6 | 0x0<br>RO | **IOCHK# NMI Source Status (IOCHK_NMI_STS):** This bit is set if an ISA/LPC agent (via SERIRQ) asserts IOCHK# and bit 3 is cleared in this register. This interrupt source is enabled by setting bit 3 to 0.<br>To reset the interrupt, set bit 3 to 1 and then set it to 0.<br>This bit is read-only. When writing to port 61h, this bit must be a 0.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RO | **Timer Counter 2 OUT Status (TMR2_OUT_STS):** This bit reflects the current state of the 8254 Counter 2 output. Counter 2 must be programmed following any PCI reset for this bit to have a determinate value.<br>When writing to port 61h, this bit must be a 0.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO | **Reserved (RSVD):** Will return 0 when read. Writes to this bit will be ignored.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **IOCHK# NMI Enable (IOCHK_NMI_EN):**<br>• When this bit is a 1, IOCHK# NMIs are disabled and cleared.<br>• When this bit is a 0, IOCHK# NMIs are enabled.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **PCI SERR# Enable (PCI_SERR_EN):**<br>• When this bit is a 1, the SERR# NMIs are disabled and cleared.<br>• When this bit is a 0, SERR# NMIs are enabled.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Reserved (RSVD):** Unused<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Timer Counter 2 Enable (TIM_CNT2_EN):**<br>When this bit is a 0, Counter 2 counting is disabled.<br>Counting is enabled when this bit is 1.<br><br>**Power Well:** Core |

§ §

# 62 Real Time Clock - Legacy

## 62.1 Introduction and Index

The host-accessible registers for the Real Time Clock are described here.

The legend for the register access codes are described in Chapter 43, "Introduction."

### 62.1.1 Host I/O Space—Fixed Address

An index of the registers that are accessible in Host I/O Space is shown here. For a description of the RTC Configuration (RC) register see "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Table 62-1. Summary of I/O Registers—Fixed Address**

| I/O Port (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 70 (also 74) | 1 | 80 | "NMI Enable (and Real Time Clock Index) (NMI_EN)—Port 70h" on page 2888 |
| 71 (also 75) | 1 | n/a | "Real Time Clock Data—Port 71h" on page 2889 |
| **If RTC Configuration (RC) bit 2 = 0** | | | |
| 72 (also 76) | 1 | 80 | (Alias) "NMI Enable (and Real Time Clock Index) (NMI_EN)—Port 70h" on page 2888 |
| 73 (also 77) | 1 | n/a | (Alias) "Real Time Clock Data—Port 71h" on page 2889 |
| **If RTC Configuration (RC) bit 2 = 1** | | | |
| 72 (also 76) | 1 | 0 | "Real Time Clock Index to Extended RAM—Port 72h" on page 2889 |
| 73 (also 77) | 1 | n/a | "Real Time Clock Data—Port 73h" on page 2889 |

### 62.1.2 Host I/O Space—RTC Standard Static RAM Bank (Indexed Data)

An index of the registers that are accessible in Host I/O Space is shown here. The Index Value is provided by the RTC Standard RAM Index Register. If the Upper 128 Byte Enable (UE) bit is = 0, the Index value is aliased to the RTC Extended RAM Index Register. For the UE bit of the RC register see "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Table 62-2. Summary of RTC Standard Static RAM Bank — Indexed Data**

| Index Value (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 | 1 | 00 | "Seconds (Sec)—Index 0h" on page 2890 |
| 1 | 1 | 00 | "Seconds Alarm (Sec_Alarm)—Index 1h" on page 2890 |
| 2 | 1 | 00 | "Minutes—Index 2h" on page 2890 |
| 3 | 1 | 00 | "Minutes Alarm (Minutes_Alarm)—Index 3h" on page 2891 |
| 4 | 1 | 00 | "Hours—Index 4h" on page 2891 |
| 5 | 1 | 00 | "Hours Alarm (Hours_Alarm)—Index 5h" on page 2891 |
| 6 | 1 | 00 | "Day of Week (Day_of_Week)—Index 6h" on page 2892 |
| 7 | 1 | 00 | "Day of Month (Day_of_Month)—Index 7h" on page 2892 |
| 8 | 1 | 00 | "Month—Index 8h" on page 2892 |
| 9 | 1 | 00 | "Year—Index 9h" on page 2893 |
| A | 1 | 20 | "Register A (Register_A)—Index Ah" on page 2894 |
| B | 1 | 80 | "Register B - General Configuration (Register_B)—Index Bh" on page 2895 |
| C | 1 | 00 | "Register C - Flag Register (Register_C)—Index Ch" on page 2897 |
| D | 1 | 80 | "Register D - Flag Register (Register_D)—Index Dh" on page 2898 |
| E through 7F | 1 (each access) | n/a | "User RAM—Index Eh through 7Fh" on page 2898 |

### 62.1.3 Host I/O Space—RTC Extended Static RAM Bank (Indexed Data)

An index of the registers that are accessible in Host I/O Space is shown here. The Index Value is provided by the RTC Extended RAM Index Register if the Upper 128 Byte Enable (UE) bit is = 1. For the UE bit of the RC register see "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Table 62-3. Summary of RTC Extended Static RAM Bank — Indexed Data**

| Index Value (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 0 through 7F | 1 (each access) | n/a | "User RAM—Index 0h through 7Fh" on page 2899 |

### 62.1.4 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 62-4. Summary of Sideband Registers—0xD1**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 3400 | 00000000 | "RTC Configuration (RC)—Offset 3400h" on page 2900 |
| 3414 | 00 | "Backed Up Control (BUC)—Offset 3414h" on page 2901 |
| 3F04 | 00000000 | "RTC Update In Progress SMI Control (UIPSMI)—Offset 3F04h" on page 2902 |
| 3418 | 00000000 | "RTC Dynamic Clock Gating Control (RTCDCG)—Offset 3418h" on page 2903 |

## 62.2 Registers in I/O Space—Fixed Addresses

These registers are accessed in Host I/O Space through fixed I/O addresses.

### 62.2.1 NMI Enable (and Real Time Clock Index) (NMI_EN)—Port 70h

This register is write-only for normal operation. In Alternate-Access mode, this register can be read to find the NMI Enable status and the RTC index value.

Alternate-Access Mode is enabled by setting Alternate Access Mode Enable (AME), bit 17 of the 32-bit General Interrupt Control (GIC) register in relocatable Memory Space at offset D0_31FCh from SBREG_BAR.

When the Upper 128 Byte Enable (UE) bit of the RTC Configuration (RC) Register = 0, The NMI_EN register is aliased to I/O Ports 72h and 76h. See "RTC Configuration (RC)—Offset 3400h" on page 2900.

I/O Writes to 72h, 74h, and 76h do not affect the NMI Enable bit (bit 7 of 70h).

**Type:** I/O Register (Size: 8 bits)

**I/O Port:** 70h (aliased 74h. If UE = 0, also aliased to 72h and 76h)

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW/V | **NMI_EN# (NMI_EN):**<br>When this bit is a 1, all NMI sources are disabled.<br>When this bit is a 0, NMI sources are enabled.<br>I/O Writes to 72h, 74h, and 76h do not affect the state of NMI_EN. |
| 6:0 | 0x0<br>RW/V | **Real Time Clock Index (Address) (RTC_INDX):** This data goes to the RTC to select which register or CMOS RAM address is being accessed.<br><br>**Power Well:** RTC |

## 62.2.2 Real Time Clock Data—Port 71h

RTC data located at index address defined by RTC_INDX.

When the Upper 128 Byte Enable (UE) bit of the RTC Configuration (RC) Register = 0, The NMI_EN register is aliased to I/O Ports 73h and 77h. See "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 71h (aliased 75h. If UE = 0, also aliased to 73h and 77h)

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Real Time Clock Data:** This is the data located at RTC_INDX. The data is either an RTC register or CMOS RAM.<br><br>**Power Well:** RTC |

## 62.2.3 Real Time Clock Index to Extended RAM—Port 72h

This register is accessible only when the Upper 128 Byte Enable (UE) bit of the RTC Configuration (RC) Register = 1. If UE = 0, see "NMI Enable (and Real Time Clock Index) (NMI_EN)—Port 70h" on page 2888. For the RC register see "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 72h (aliased 76h)

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved:** Reserved. |
| 6:0 | 0x0<br>RW | **Real Time Clock Extended Index (Address)** This data goes to the RTC to select which Extended CMOS RAM address is being accessed.<br><br>**Power Well:** RTC |

## 62.2.4 Real Time Clock Data—Port 73h

RTC data located at index address defined by the Real Time Clock Extended Index register. This register is accessible only when the Upper 128 Byte Enable (UE) bit of the RTC Configuration (RC) Register = 1. If UE = 0, see "Real Time Clock Data—Port 71h" on page 2889. For the RC register see "RTC Configuration (RC)—Offset 3400h" on page 2900.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port:** 73h (aliased 77h)

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Real Time Clock Data:** This is the data located at the Extended RAM address defined by the Real Time Clock Extended Index register. The data is Extended CMOS RAM.<br><br>**Power Well:** RTC |

## 62.3 Registers in Host I/O Space—RTC Standard Static RAM Bank (Indexed Data)

### 62.3.1 Seconds (Sec)—Index 0h

RTC Index: 00h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time second.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Seconds (Sec):** Time Seconds<br><br>**Power Well:** RTC |

### 62.3.2 Seconds Alarm (Sec_Alarm)—Index 1h

RTC Index: 01h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Seconds Alarm.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 1h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Seconds Alarm (Sec_Alarm):** Sec_Alarm<br><br>**Power Well:** RTC |

### 62.3.3 Minutes—Index 2h

RTC Index: 02h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time Minutes.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 2h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Minutes:** Time Minutes<br><br>**Power Well:** RTC |

### 62.3.4 Minutes Alarm (Minutes_Alarm)—Index 3h

RTC Index: 03h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Alarm Minutes.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 3h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Minutes Alarm (Minutes_Alarm):** Alarm Minutes<br><br>**Power Well:** RTC |

### 62.3.5 Hours—Index 4h

RTC Index: 04h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time Hours.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 4h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Hours:** Time Hours<br><br>**Power Well:** RTC |

### 62.3.6 Hours Alarm (Hours_Alarm)—Index 5h

RTC Index: 05h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Alarm Hours.

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** 5h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Hours Alarm (Hours_Alarm):** Alarm Hours<br><br>**Power Well:** RTC |

## 62.3.7 Day of Week (Day_of_Week)—Index 6h

RTC Index: 06h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Day of Week.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** 6h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Day of Week (Day_of_Week):** This field indicates current Day of Week<br><br>**Power Well:** RTC |

## 62.3.8 Day of Month (Day_of_Month)—Index 7h

RTC Index: 07h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Day of Month.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** 7h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Day of Month (Day_of_Month):** This field indicates current Day of Month<br><br>**Power Well:** RTC |

## 62.3.9 Month—Index 8h

RTC Index: 08h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Month.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** 8h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Month:** This field indicates current Month<br><br>**Power Well:** RTC |

## 62.3.10 Year—Index 9h

RTC Index: 09h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Year.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** 9h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Year:** This field indicates current Year<br><br>**Power Well:** RTC |

## 62.3.11 Register A (Register_A)—Index Ah

RTC Index: 0Ah Attribute: Read/Write Default Value: Undefined Size: 8-bit Lockable: No Power Well: RTC This register is used for general configuration of the RTC functions. None of the bits are affected by RSMRST# or any other reset signal.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** Ah

**Default:** 20h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **UPDATE IN PROGRESS (UIP):** This bit may be monitored as a status flag.<br>When asserted as a 1, the update is soon to occur or is in progress.<br>If 0, the update cycle will not start for at least 488 s. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0. The ICH provides the ability to generate SMI# based on either the 0-to-1 or 1-to-0 transition of this bit. This can be useful for work-arounds and debug in silicon.<br><br>**Power Well:** RTC |
| 6:4 | 0x2<br>RW | **Division Chain Select (DV_2_0):** These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV(2) corresponds to bit 6.<br>• 010 Normal Operation<br>• 11X Divider Reset<br>• 101 Bypass 15 stages (test mode only)<br>• 100 Bypass 10 stages (test mode only)<br>• 011 Bypass 5 stages (test mode only)<br>• 001 Invalid<br>• 000 Invalid<br><br>**Power Well:** RTC |
| 3:0 | 0x0<br>RW | **Rate Select (RS_3_0):** Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in Register B. Otherwise this tap will set the PF flag of Register C. If the periodic interrupt is not to be used, these bits should all be set to zero. RS3 corresponds to bit 3. The Periodic Rate is as follows:<br>• 0000 Interrupt never toggles<br>• 0001 3.90625 ms<br>• 0010 7.8125 ms<br>• 0011 122.070 s<br>• 0100 244.141 s<br>• 0101 488.281 s<br>• 0110 976.5625 s<br>• 0111 1.953125 ms<br>• 1000 3.90625 ms<br>• 1001 7.8125 ms<br>• 1010 15.625 ms<br>• 1011 31.25 ms<br>• 1100 62.5 ms<br>• 1101 125 ms<br>• 1110 250 ms<br>• 1111 500 ms<br><br>**Power Well:** RTC |

## 62.3.12 Register B - General Configuration (Register_B)—Index Bh

RTC Index: 0Bh Attribute: Read/Write Default Value: U0U00UUU Size: 8-bit Lockable: No Power Well: RTC

**Type:** RTC Standard Static RAM (Size: 8 bits) **Index:** Bh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Update Cycle Inhibit (SET):** Enables/Inhibits the update cycles.<br>When SET is 0, update cycle occurs normally once each second.<br>If set to one, a current update cycle will abort and subsequent update cycles will not occur until SET is returned to zero.<br>When set is one, the BIOS may initialize time and calendar bytes safely.<br>This bit is not affected by RSMRST# nor any other reset signal.<br>***Note:*** Software must ensure this bit is at least transitioned from '1' to '0' once whenever the RTC coin battery is inserted. This is to ensure that the internal RTC time updates occur properly.<br><br>**Power Well:** RTC |
| 6 | 0x0<br>RW | **Periodic Interrupt Enable (PIE):** If set to 1, the Periodic Interrupt Enable (PIE) bit allows an interrupt to occur with a time base set with the RS bits of register A.<br>This bit is cleared by RSMRST#, but not on any other reset<br><br>**Power Well:** RTC |
| 5 | 0x0<br>RW | **Alarm Interrupt Enable (AIE):** If set to one, the Alarm Interrupt Enable (AIE) bit allows an interrupt to occur when the AF is one as set from an alarm match from the update cycle.<br>An alarm can occur once a second, one an hour, once a day, or once a month.<br>This bit is cleared by RTEST# #, but not on any other reset<br><br>**Power Well:** RTC |
| 4 | 0x0<br>RW | **Update-ended Interrupt Enable: (UIE):** If set to one, the Update-ended Interrupt Enable (UIE) bit allows an interrupt to occur when the update cycle ends.<br>This bit is cleared by RSMRST#, but not on any other reset<br><br>**Power Well:** RTC |
| 3 | 0x0<br>RW | **Square Wave Enable (SQWE):** The Square Wave Enable bit serves no function in this device, yet is left in this register bank to provide compatibility with the Motorola 146818B.<br>There is not SQW pin on this device.<br>This bit is cleared by RSMRST#, but not on any other reset.<br><br>**Power Well:** RTC |
| 2 | 0x0<br>RW | **Data Mode (DM):** The Data Mode (DM) bit specifies either binary or BCD data representation.<br>A one denotes binary, and zero denotes BCD.<br>This bit is not affected by RSMRST# nor any other reset signal.<br><br>**Power Well:** RTC |
| 1 | 0x0<br>RW | **Hour Format (HOURFORM):** This bit indicates the hour byte format.<br>If one, twenty-four hour mode is selected.<br>If zero, twelve-hour mode is selected. In twelve hour mode, the seventh bit represents AM as zero and PM as one.<br>This bit is not affected by RSMRST# nor any other reset signal.<br><br>**Power Well:** RTC |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 0 | 0x0<br>RW | **Daylight Savings Enable (DSE):** The Daylight Savings Enable bit triggers two special hour updates per year when set to one.<br>One is on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM.<br>The other is the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM.<br>The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly.<br>These special update conditions do not occur when the DSE bit is set to zero.<br>The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years.<br>This bit is not affected by RSMRST# nor any other reset signal.<br>If BUC.DSO bit is set, the DSE bit continues to be a R/W bit, but Daylight Saving is disabled regardless of the DSE bit.<br><br>**Power Well:** RTC |

## 62.3.13 Register C - Flag Register (Register_C)—Index Ch

RTC Index: 0Ch Attribute: Read-Only (Writes have no effect). Default Value: 00U00000 Size: 8-bit Lockable: No Power Well: RTC

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** Ch

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Interrupt Request Flag (IRQF):** Interrupt Request Flag = PF * PIE + AF * AIE + UF * UFE.<br>This also causes the RTC Interrupt to be asserted.<br>This bit is cleared upon RSMRST# or a read of Register C.<br><br>**Power Well:** RTC |
| 6 | 0x0<br>RO | **Periodic Interrupt Flag (PF):** Periodic interrupt Flag will be one when the tap as specified by the RS bits of register A is one.<br>If no taps are specified, this flag bit will remain at zero.<br>This bit is cleared upon RSMRST# or a read of Register C.<br><br>**Power Well:** RTC |
| 5 | 0x0<br>RO | **Alarm Flag (AF):** Alarm Flag will be high after all Alarm values match the current time.<br>This bit is cleared upon RTEST# or a read of Register C.<br><br>**Power Well:** RTC |
| 4 | 0x0<br>RO | **Update-ended Flag (UF):** Updated-ended flag will be high immediately following an update cycle for each second.<br>The bit is cleared upon RSMRST# or a read of Register C.<br><br>**Power Well:** RTC |
| 3:0 | 0x0<br>RO | **Reserved (RSVD):** Will always report 0<br><br>**Power Well:** RTC |

## 62.3.14 Register D - Flag Register (Register_D)—Index Dh

RTC Index: 0Dh Attribute: Read/Write Default Value: 10UUUUUU Size: 8-bit Lockable: No Power Well: RTC

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** Dh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Valid RAM and Time Bit (VRT):** This bit is hard-wired to 1 in the RTC power well.<br>This bit should always be written as a 0 for write cycle, however it will return a 1 for read cycles.<br><br>**Power Well:** RTC |
| 6 | 0x0<br>RW | **Reserved (RSVD):** This bit always returns a 0 and should be set to 0 for write cycles.<br><br>**Power Well:** RTC |
| 5:0 | 0x0<br>RW | **Date Alarm (Date_Alarm):** These bits store the date of month alarm value.<br>If set to 000000, then a dont care state is assumed.<br>The host must configure the dates alarm for these bits to do anything, yet they can be written at any time.<br>If the date alarm is not enabled, these bits will return zeros to mimic the functionality of the Motorola 146818B.<br>These bits are not affected by any reset assertion.<br><br>**Power Well:** RTC |

## 62.3.15 User RAM—Index Eh through 7Fh

Access to 114 bytes of user Static RAM located in the RTC Power Well of the SoC.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** Dh through 7Fh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | n/a<br>RW | **User RAM Data:**<br><br>**Power Well:** RTC |

## 62.4 Registers in Host I/O Space—RTC Extended Static RAM Bank (Indexed Data)

### 62.4.1 User RAM—Index 0h through 7Fh

Access to 128 bytes of user Static RAM located in the RTC Power Well of the SoC.

**Type:** RTC Standard Static RAM (Size: 8 bits)

**Index:** 0h through 7Fh

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | n/a<br>RW | **User RAM Data:**<br><br>**Power Well:** RTC |

## 62.5 Sideband Registers

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 62.5.1 RTC Configuration (RC)—Offset 3400h

All bits in this register are in the Primary Well and cleared by host_side_rst_b.

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD1] + 3400h
**Host Memory Space:** SBREG_BAR + 0xD10000 + 3400h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1L | **Bios Interface Lock-Down (BILD):** When set, prevents RTC version of TS (BUC.TS) from being changed.<br>This bit can only be written from 0 to 1 once.<br>This BILD bit has different function compared to the LPC and SPI version but BIOS should set all the corresponding bits after reset in order to lock down the BIOS interface correctly.<br><br>**Power Well:** Core |
| 30:7 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 6 | 0x0<br>RW | **RTC High Power Mode HW Disable (HPM_HW_DIS):**<br>When set to 1 the internal VRM that generates the rtc well supply voltage in SUS mode is disabled when SLP_S0# is asserted to '0'. (via irtcdswen pin to RTC EBB).<br>When 0, HW control of the RTC internal VRM is disabled.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **RTC High Power Mode SW Disable (HPM_SW_DIS):**<br>When set to 1 the internal VRM that generates the rtc well supply voltage in SUS mode is disabled (via irtcdswen pin to RTC EBB).<br>When 0 the internal VRM powers the rtc well when RSMRST# is '1'. (default)<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW/1L | **Upper 128 Byte Lock (UL):** When set, bytes 38h-3Fh in the upper 128 byte bank of RTC RAM are locked and cannot be accessed.<br>Writes will be dropped and reads will not return any guaranteed data. Bit reset on system reset.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW/1L | **Lower 128 Byte Lock (LL):** When set, bytes 38h-3Fh in the lower 128 byte bank of RTC RAM are locked and cannot be accessed.<br>Writes will be dropped and reads will not return any guaranteed data. Bit reset on system reset.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **Upper 128 Byte Enable (UE):** When set, the upper 128 byte bank of RTC RAM can be accessed.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |

## 62.5.2 Backed Up Control (BUC)—Offset 3414h

All bits in this register are in the RTC well and only cleared by RTEST.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD1] + 3414h
**Host Memory Space:** SBREG_BAR + 0xD10000 + 3414h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** RTC |
| 6 | 0x0<br>RW | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** RTC |
| 5 | 0x0<br>RW | **LAN Disable (LanDisable):**<br>• '0': LAN is Enabled '1': LAN is Disabled. It will not claim any configuration cycles on SPXB or MBB nor initiate any cycles on SPXB or MBB. When the RTC backup copy of the LAN Disable Fuse is set, this register is Read Only.<br>• '1'. If the RTC backup copy of the LAN Disable Fuse is not set, this register is reset to 0 on RTEST#. If the Function Disable SUS Well Lockdown register is set, this bit can not be changed by software.<br><br>**Power Well:** RTC |
| 4 | 0x0<br>RW | **Daylight Savings Override (SDO):**<br>• When this bit is a '1', the DSE bit in the RTC Register B bit(0) is a RW bit but has no effect where daylight savings is hard-disabled internally.<br>• When this bit is a '0', the DSE bit in the RTC register B bit(0) is a RW bit that is configurable by software to enable the daylight savings. System BIOS shall configure this bit accordingly during the boot process before RTC time is initialized.<br><br>**Power Well:** RTC |
| 3 | 0x0<br>RW | **NetDetect Enable (NDE):**<br>• If this bit is '1' and the South MLink Enable bit is '0' (ME PM RTCPMCFG.SMLEN), then the GPIO(14) input signal is muxed onto the South MLink MLCLK pin as a NetDetect Request signal to the wireless LAN component.<br>• If the South MLink Enable bit is a '1', then the South MLink MLCLK pin is used as MLCLK, independent of the value of this register.<br>• If both the South MLink Enable bit is '0' and this NetDetect Enable bit is '0', then the South MLink MLCLK pin is tri-stated. This register is in the RTC well instead of the SUS well to maintain state if the SUS well power is removed in S4.<br><br>**Power Well:** RTC |
| 2 | 0x0<br>RW | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** RTC |
| 1 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved<br><br>**Power Well:** RTC |
| 0 | 0x0<br>RW | **Top Swap (TS):** This should be set by BIOS when the corresponding TS bit in the LPC controller is set in order to properly restore the state of that field after reset since they are not preserved in an RTC well bit in those devices.<br>If SoC is strapped for Top-Swap (GNT(3)# is low at rising edge of PWROK), then this bit cannot be cleared by software.<br>The strap jumper should be removed and the system rebooted.<br><br>**Power Well:** RTC |

### 62.5.3 RTC Update In Progress SMI Control (UIPSMI)—Offset 3F04h

This register exists in the RTC power well.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD1] + 3F04h
**Host Memory Space:** SBREG_BAR + 0xD10000 + 3F04h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW/1C | **RTC UIP Low-to-High (UIP_L2H):** This sticky status bit is set whenever the RTC Update-In-Progress signal transitions from low to high (i.e., at the start of an update).<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW/1C | **RTC UIP High-to-Low (UIP_H2L):** This sticky status bit is set whenever the RTC Update-In-Progress signal transitions from high to low (i.e., at the start of an update).<br><br>**Power Well:** Core |
| 15:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **RTC UIP Low-to-High SMI Enable (UIP_L2H_SMI_en):** When this bit is set, a '1' in bit 17 will assert the internal SMI signal to the Power Management SMI logic<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **RTC UIP High-to-Low SMI Enable (UIP_H2L_SMI_en):** When this bit is set, a '1' in bit 16 will assert the internal SMI signal to the Power Management SMI logic<br><br>**Power Well:** Core |

### 62.5.4 RTC Dynamic Clock Gating Control (RTCDCG)—Offset 3418h

All bits in this register are in the Primary Well and cleared by host_side_rst_b.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD1] + 3418h
**Host Memory Space:** SBREG_BAR + 0xD10000 + 3418h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** RTC |
| 7:3 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved for future use<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **pgcb_clk (12MHz) Dynamic Clock Gate Enable (RTCPGCBDCGEN):**<br>• 0 - Disable dynamic clock gate on pgcb_clk (Default)<br>• 1 - Enable dynamic clock gate on pgcb_clk<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **ipciclk_clk (24MHz) Dynamic Clock Gate Enable (RTCPCICLKDCGEN):**<br>• 0 - Disable dynamic clock gate on ipci_clk (Default)<br>• 1 - Enable dynamic clock gate on ipci_clk<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **rosc_side_clk (120MHz) Dynamic Clock Gate Enable (RTCROSIDEDCGEN):**<br>• 0 - Disable dynamic clock gate on rosc_side_clk (Default)<br>• 1 - Enable dynamic clock gate on rosc_side_clk<br><br>**Power Well:** Core |

§ §

# 63 High Precision Event Timer - Legacy

## 63.1 Introduction and Index

The host-accessible registers for the High Precision Event Timer are described here.

The legend for the register access codes are described in Chapter 43, "Introduction."

### 63.1.1 Host Memory Space—Fixed Address

An index of the registers that are accessible in Host Memory Space is shown here.

**Table 63-1. Summary of Memory Mapped I/O Registers—0xFED00000**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 10 | 8 | 0000000000000000 | "General Config Register (GEN_CFG)—Offset 10h" on page 2906 |
| 20 | 8 | 0000000000000000 | "General Interrupt Status Register (GEN_INT_STS)—Offset 20h" on page 2907 |

## 63.2 Registers in Memory Space—Fixed Addresses

These registers are accessed in Host Memory Space through fixed memory addresses.

### 63.2.1 General Config Register (GEN_CFG)—Offset 10h

Software can read the various bytes in this register using 32-bit or 64-bit accesses. 32-bit accesses may only be done to offset 010h or 014h. 64-bit accesses may only be done to 010h.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Address:** [0xFED00000] + 10h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:2 | 0x0<br>RO | **Reserved (RSV_63_2):** These bits will return 0 when read. Writes will have no effect.<br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Legacy Rout (LEG_RT_CNF)**<br>If the LEG_RT_CNF bit is set, then the interrupts will be routed as follows:<br>• **Timer 0** will be routed to IRQ0 in 8259 or IRQ2 in the I/O APIC<br>• **Timer 1** will be routed to IRQ8 in 8259 or IRQ8 in the I/O APIC<br>• **Timer 2-7** will be routed as per the routing in that timer config register.<br>If the LEG_RT_CNF bit is set, the individual routing bits for timers 0 and 1 (APIC or FSB) will have no impact.<br>Otherwise, if the LEG_RT_CNF bit is not set, the individual routing bits for each of the timers are used.<br>**Power Well:** Core |
| 0 | 0x0<br>RW | **Overall Enable (ENABLE_CNF):** This bit must be set to enable any of the timers to generate interrupts.<br>If this bit is 0, then the main counter will halt (will not increment) and no interrupts will be caused by any of these timers.<br>This bit will default to 0. BIOS can set it to 1 or 0.<br>**Power Well:** Core |

### 63.2.2 General Interrupt Status Register (GEN_INT_STS)—Offset 20h

Software can read the various bytes in this register using 32-bit or 64-bit accesses. 32-bit accesses may only be done to offset 020h or 024h. 64-bit accesses may only be done to 020h.

**Type:** Memory Mapped I/O Register (Size: 64 bits)

**Address:** [0xFED00000] + 20h

**Default:** 0000000000000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:8 | 0x0<br>RO | **Reserved (RSV):** These bits will return 0 when read. Writes will have no effect.<br>**Power Well:** Core |
| 7 | 0x0<br>RW/1C | **Timer 7 Interrupt Active (T07_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 6 | 0x0<br>RW/1C | **Timer 6 Interrupt Active (T06_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 5 | 0x0<br>RW/1C | **Timer 5 Interrupt Active (T05_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 4 | 0x0<br>RW/1C | **Timer 4 Interrupt Active (T04_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 3 | 0x0<br>RW/1C | T**imer 3 Interrupt Active (T03_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 2 | 0x0<br>RW/1C | **Timer 2 Interrupt Active (T02_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 1 | 0x0<br>RW/1C | **Timer 1 Interrupt Active (T01_INT_STS)**<br>Same functionality as Timer 0 Interrupt Active (T00_INT_STS).<br>**Power Well:** Core |
| 0 | 0x0<br>RW/1C | **Timer 0 Interrupt Active (T00_INT_STS)**<br>The functionality of this bit depends on whether the edge or level-triggered mode is used for this timer:<br>• If set to edge triggered mode: This bit will always read as 0 and writes will have no effect.<br>• If set to level-triggered mode: This bit will be set by hardware if the corresponding timer interrupt is active. Once the bit is set, it can be cleared by software writing a 1 to the same bit position. Writes of 0 to this bit will have no effect. For example if the bit is already set, a write of 0 will not clear the bit.<br>This bit defaults to 0.<br>**Power Well:** Core |

§ §

# 64 PIC and I/O APIC Controllers

## 64.1 Introduction and Index

The host-accessible registers for the PIC and I/O APIC Controller are described here.

The legend for the register access codes are described in Chapter 43, "Introduction."

### 64.1.1 Host Configuration Space

These registers are discovered in Host Configuration Space starting at Bus 0, Device 31 (decimal), Function 1. The offset addresses are listed.

**Table 64-1. Summary of PCI Configuration Registers—0/31/1**

| Offset (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 64 | 2 | 0000 | "IOxAPIC Configuration (IOAC)—Offset 64h" on page 2913 |

### 64.1.2 Host Memory Space—Fixed Address

An index of the registers that are accessible in Host Memory Space is shown here.

The IOxAPIC Interrupt Message table is located in Host Memory Space starting at address FEE0_0000h. Construction of the address for a table entry is shown in Table 64-2.

**Table 64-2. IOxAPIC Interrupt Message Address**

| Address Bit | Short Description | Description |
|---|---|---|
| 31:20 | FEEh | Hardwired as FEEh |
| 19:12 | Destination ID | Destination ID of the local APIC - from DID field of the interrupt RTE |
| 11:4 | Extended Destination ID | Extended destination ID of the local APIC - from EDID field of the interrupt RTE |
| 3 | Redirection Hint | Redirection Hint - '1" If the DLM field = “Lowest Priority” (001) for the interrupt RTE. Otherwise '0' |
| 2 | Destination Mode | Used by the Local APIC to determine whether it is the destination of the Interrupt Message |
| 1:0 | 00 | Hardwired as 00 - Interrupt Message data is written as a Double Word (32 bits) |

For the description of an IOxAPIC Interrupt Message, see Section 64.3.3, “IOxAPIC Interrupt Messages for Local APIC—FEE0_0000h” on page 2914.

Also located in Host Memory Space are the 32-bit IOxAPIC Index (IDX), Window (WDW), and End-of-Interrupt (EOI) registers. See Table 64-3, “Summary of Memory-Mapped Registers—Fixed Addresses” on page 2910.

For the information accessible through the IOxAPIC Index/Window mechanism, see Table 64-4, “Summary of IOxAPIC Index/Window-Accessed Registers” on page 2911.

**Table 64-3. Summary of Memory-Mapped Registers—Fixed Addresses**

| Address (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| FECO_0000 | 4 | 00000000 | “Index Register (IDX)—FEC0_0000h” on page 2914 |
| FECO_0010 | 4 | 00000000 | “Window Register (WDW)—FEC0_0010h” on page 2914 |

## Table 64-4. Summary of IOxAPIC Index/Window-Accessed Registers

| Index (IDX) Value (hex) | Double-Word Accesses | Default Value (hex) | Contents of Window (WDW) Register Name (Register Symbol) |
|---|---|---|---|
| 1 | 1 | 00770020 | "Version (VS)—Offset 1h" on page 2917 |
| 2 through F | 1 | 00000000 | Reserved |
| 10, 11 | 2 | 0000000000010000 | "Redirection Table Entry 0 (RTE0)—Offset 10h" on page 2918 |
| 12, 13 | 2 | 0000000000010000 | "Redirection Table Entry 1 (RTE1)—Offset 12h" on page 2920 |
| 14, 15 | 2 | 0000000000010000 | "Redirection Table Entry 2 (RTE2)—Offset 14h" on page 2922 |
| through | ... | ... | ... |
| FC, FD | 2 | 0000000000010000 | "Redirection Table Entry 118 (RTE118)—Offset FCh" on page 2924 |
| FE, FF | 2 | 0000000000010000 | "Redirection Table Entry 119 (RTE119)—Offset FEh" on page 2926 |

### 64.1.3 Host I/O Space—Fixed Address

An index of the registers that are accessible in Host I/O Space is shown here.

**Table 64-5. Summary of I/O Registers—Fixed Location**

| I/O Port Address (hex) | Bytes | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 20 | 1 | 11 | "Master Initialization Command Word 1 (MICW1)—I/O Port 20h" on page 2928 |
| 21 | 1 | 00 | "Master Initialization Command Word 2 (MICW2)—I/O Port 21h" on page 2929 |
| 21 | 1 | 07 | "Master Initialization Command Word 3 (MICW3)—I/O Port 21h" on page 2929 |
| 21 | 1 | 00 | "Master Initialization Command Word 4 (MICW4)—I/O Port 21h" on page 2930 |
| 21 | 1 | 00 | "Master Operational Control Word 1 (MOCW1)—I/O Port 21h" on page 2930 |
| 20 | 1 | 00 | "Master Operational Control Word 2 (MOCW2)—I/O Port 20h" on page 2931 |
| 20 | 1 | 08 | "Master Operational Control Word 3 (MOCW3)—I/O Port 20h" on page 2932 |
| A0 | 1 | 11 | "Slave Initialization Command Word 1 (SICW1)—I/O Port A0h" on page 2933 |
| A1 | 1 | 00 | "Slave Initialization Command Word 2 (SICW2)—I/O Port A1h" on page 2934 |
| A1 | 1 | 07 | "Slave Initialization Command Word 3 (SICW3)—I/O Port A1h" on page 2934 |
| A1 | 1 | 00 | "Slave Initialization Command Word 4 (SICW4)—I/O Port A1h" on page 2935 |
| A1 | 1 | 00 | "Slave Operational Control Word 1 (SOCW1)—I/O Port A1h" on page 2935 |
| A0 | 1 | 00 | "Slave Operational Control Word 2 (SOCW2)—I/O Port A0h" on page 2936 |
| A0 | 1 | 08 | "Slave Operational Control Word 3 (SOCW3)—I/O Port A0h" on page 2937 |
| 4D0 | 1 | 00 | "Master Edge/Level Control (ELCR1)—I/O Port 4D0h" on page 2938 |
| 4D1 | 1 | 00 | "Slave Edge/Level Control (ELCR2)—I/O Port 4D1h" on page 2938 |

## 64.2 Registers in Configuration Space

### 64.2.1 IOxAPIC Configuration (IOAC)—Offset 64h

**Type:** PCI Configuration Register (Size: 16 bits)

**Offset:** [B:0, D:31, F:1] + 64h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:9 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 8 | 0x0<br>RW | **Address Enable (AE):** When set, the P2SB will decode the IOxAPIC Host Root memory-address range selected by bits 7:0 below.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **APIC Range Select (ASEL):** These eight bits define the eight address bits 19:12 for the IOxAPIC range.<br>The default value of 00h enables compatibility with prior Intel products as an initial value. This value must not be changed by software unless the IOxAPIC Address Enable (AE) bit is cleared.<br><br>**Power Well:** Core |

## 64.3 Registers in Memory Space—Fixed Addresses

These registers are accessed in Host Memory Space through fixed memory addresses.

### 64.3.1 Index Register (IDX)—FEC0_0000h

This 8-bit register selects which indirect register appears in the window register to be manipulated by software. Software will program this register to select the desired APIC internal register.

**Type:** Memory-Mapped I/O Register (Size: 32 bits)

**Address:** FEC0_0000h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:8 | 0x0<br>RO | **Reserved (RSVD):** Reserved.<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Index Register (IDX):** Index Register for APIC<br><br>**Power Well:** Core |

### 64.3.2 Window Register (WDW)—FEC0_0010h

This 32-bit register specifies the data to be read or written to the register pointed to by the IDX register. This register can be accessed only in DWord quantities.

**Type:** Memory-Mapped I/O Register (Size: 32 bits)

**Address:** FEC0_0010h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW/V | **Window Register (WDW):** Window Register for APIC<br><br>**Power Well:** Core |

### 64.3.3 IOxAPIC Interrupt Messages for Local APIC—FEE0_0000h

The 32-bit IOxAPIC Interrupt Messages sent to the Local APICs are written in Host Memory Space starting at address FEE0_0000h through address FEEF_FFFCh. Each message has the following data format.

**Type:** Memory-Mapped I/O Register (Size: 32 bits)

**Address:** FEE0_0000h through FEEF_FFCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved:** Reserved. Hardwired as 0. |
| 15 | 0x0<br>RO | **Trigger Mode:** From the TM field of the Redirection Table Entry (RTE).<br>• '0' indicates the interrupt is edge sensitive.<br>• '1' indicates the interrupt is level sensitive. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14 | 0x0<br>RO | **Delivery Status:** Is always '1' indicating an "Assert Message" cycle. |
| 13:12 | 0x0<br>RO | **Reserved:** Reserved. Hardwired as 0. |
| 11 | 0x0<br>RO | **Destination Mode:** From the DSM field of the RTE. Used by the Local APIC to determine whether it is the destination of the message. |
| 10:8 | 0x0<br>RO | **Delivery Mode:** From the DLM field of the RTE. Establishes how the APICs listed in the destination field should act upon reception of this signal.<br>Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. |
| 7:0 | 0x0<br>RO | **Vector:** From the VCT field of the RTE. Defines the Interrupt Vector for this interrupt.<br>Values range between 10h and FEh. |

## 64.4 Registers Accessed through IOxAPIC IDX and WDW

These registers are accessed through the IOxAPIC Index/Window mechanism. For each Index (IDX) value, one double-word is accessible in the Window (WDW) register.

### 64.4.1 Version (VS)—Offset 1h

**Type:** IOxAPIC IDX/WDW (Size: 32 bits)

**IDX:** 1h

**Default:** 00770020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 23:16 | 0x77<br>RW/O | **Maximum Redirection Entries (MRE):** This is the entry number (0 being the lowest entry) of the highest entry in the redirection table.<br>The default value is 77h to indicate a maximum of 120 (decimal) Redirection Table Entries.<br>This field is Read-Write-Once. BIOS must write to this field after PLTRST# to lockdown the value.<br>This allows BIOS to utilize some of the entries for its own purpose and thus advertising fewer IOxAPIC Redirection Table Entries to the Operating System.<br>BIOS may to program this field up to 77h (maximum 120 entries).<br><br>**Power Well:** Core |
| 15 | 0x0<br>RO | **Pin Assertion Register Supported (PRQ):** The default value indicates that the IOxAPIC does not implement the Pin Assertion Register.<br><br>**Power Well:** Core |
| 14:8 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 7:0 | 0x20<br>RO | **Version (VS):** Identifies the implementation version as IOxAPIC.<br><br>**Power Well:** Core |

## 64.4.2 Redirection Table Entry 0 (RTE0)—Offset 10h

**Type:** IOxAPIC IDX/WDW (Size: 64 bits)

**IDX:** 10h, 11h

**Default:** 0000000000010000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x0<br>RW | **Destination ID (DID):** Destination ID of the local APIC<br><br>**Power Well:** Core |
| 55:48 | 0x0<br>RW | **Extended Destination ID (EDID):** Extended destination ID of the local APIC.<br><br>**Power Well:** Core |
| 47:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x1<br>RW | **Mask (MSK):**<br>When set, interrupts are not delivered nor held pending.<br>When cleared, and edge or level on this interrupt results in the delivery of the interrupt.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>When cleared, the interrupt is edge sensitive.<br>When set, the interrupt is level sensitive.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RO/V | **Remote IRR (RIRR):** This is used for level triggered interrupts - its meaning is undefined for edge triggered interrupts.<br>This bit is set when IOxAPIC sends the level interrupt message to the CPU.<br>This bit is cleared when an EOI message is received that matches the VCT field.<br>This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Polarity (POL):** This specifies the polarity of each interrupt input.<br>When cleared, the signal is active high.<br>When set, the signal is active low.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO/V | **Delivery Status (DS):** This field contains the current status of the delivery of this interrupt.<br>When set, an interrupt is pending and not yet delivered.<br>When cleared, there is no activity for this entry<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. These encodings are:<br>• 000 - Fixed<br>• 001 - Lowest Priority<br>• 010 - SMI (Not supported)<br>• 011 - Reserved<br>• 100 - NMI (Not supported)<br>• 101 - INIT (Not supported)<br>• 110 - Reserved<br>• 111 - ExtINT<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Vector (VCT):** This field contains the interrupt vector for this interrupt.<br>Values range between 10h and FEh.<br><br>**Power Well:** Core |

### 64.4.3 Redirection Table Entry 1 (RTE1)—Offset 12h

**Type:** IOxAPIC IDX/WDW (Size: 64 bits)

**IDX:** 12h, 13h

**Default:** 0000000000010000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x0<br>RW | **Destination ID (DID):** Destination ID of the local APIC<br><br>**Power Well:** Core |
| 55:48 | 0x0<br>RW | **Extended Destination ID (EDID):** Extended destination ID of the local APIC.<br><br>**Power Well:** Core |
| 47:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x1<br>RW | **Mask (MSK):**<br>When set, interrupts are not delivered nor held pending.<br>When cleared, and edge or level on this interrupt results in the delivery of the interrupt.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>When cleared, the interrupt is edge sensitive.<br>When set, the interrupt is level sensitive.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RO/V | **Remote IRR (RIRR):** This is used for level triggered interrupts - its meaning is undefined for edge triggered interrupts.<br>This bit is set when IOxAPIC sends the level interrupt message to the CPU.<br>This bit is cleared when an EOI message is received that matches the VCT field.<br>This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Polarity (POL):** This specifies the polarity of each interrupt input.<br>When cleared, the signal is active high.<br>When set, the signal is active low.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO/V | **Delivery Status (DS):** This field contains the current status of the delivery of this interrupt.<br>When set, an interrupt is pending and not yet delivered.<br>When cleared, there is no activity for this entry<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. These encodings are:<br>• 000 - Fixed<br>• 001 - Lowest Priority<br>• 010 - SMI (Not supported)<br>• 011 - Reserved<br>• 100 - NMI (Not supported)<br>• 101 - INIT (Not supported)<br>• 110 - Reserved<br>• 111 - ExtINT<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Vector (VCT):** This field contains the interrupt vector for this interrupt.<br>Values range between 10h and FEh.<br><br>**Power Well:** Core |

### 64.4.4 Redirection Table Entry 2 (RTE2)—Offset 14h

**Type:** IOxAPIC IDX/WDW (Size: 64 bits)

**IDX:** 14h, 15h

**Default:** 0000000000010000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x0<br>RW | **Destination ID (DID):** Destination ID of the local APIC<br><br>**Power Well:** Core |
| 55:48 | 0x0<br>RW | **Extended Destination ID (EDID):** Extended destination ID of the local APIC.<br><br>**Power Well:** Core |
| 47:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x1<br>RW | **Mask (MSK):**<br>When set, interrupts are not delivered nor held pending.<br>When cleared, and edge or level on this interrupt results in the delivery of the interrupt.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>When cleared, the interrupt is edge sensitive.<br>When set, the interrupt is level sensitive.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RO/V | **Remote IRR (RIRR):** This is used for level triggered interrupts - its meaning is undefined for edge triggered interrupts.<br>This bit is set when IOxAPIC sends the level interrupt message to the CPU.<br>This bit is cleared when an EOI message is received that matches the VCT field.<br>This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Polarity (POL):** This specifies the polarity of each interrupt input.<br>When cleared, the signal is active high.<br>When set, the signal is active low.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO/V | **Delivery Status (DS):** This field contains the current status of the delivery of this interrupt.<br>When set, an interrupt is pending and not yet delivered.<br>When cleared, there is no activity for this entry<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. These encodings are:<br>• 000 - Fixed<br>• 001 - Lowest Priority<br>• 010 - SMI (Not supported)<br>• 011 - Reserved<br>• 100 - NMI (Not supported)<br>• 101 - INIT (Not supported)<br>• 110 - Reserved<br>• 111 - ExtINT<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Vector (VCT):** This field contains the interrupt vector for this interrupt.<br>Values range between 10h and FEh.<br><br>**Power Well:** Core |

## 64.4.5 Redirection Table Entry 118 (RTE118)—Offset FCh

**Type:** IOxAPIC IDX/WDW (Size: 64 bits)

**IDX:** FCh, FDh

**Default:** 0000000000010000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x0<br>RW | **Destination ID (DID):** Destination ID of the local APIC<br><br>**Power Well:** Core |
| 55:48 | 0x0<br>RW | **Extended Destination ID (EDID):** Extended destination ID of the local APIC.<br><br>**Power Well:** Core |
| 47:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x1<br>RW | **Mask (MSK):**<br>When set, interrupts are not delivered nor held pending.<br>When cleared, and edge or level on this interrupt results in the delivery of the interrupt.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>When cleared, the interrupt is edge sensitive.<br>When set, the interrupt is level sensitive.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RO/V | **Remote IRR (RIRR):** This is used for level triggered interrupts - its meaning is undefined for edge triggered interrupts.<br>This bit is set when IOxAPIC sends the level interrupt message to the CPU.<br>This bit is cleared when an EOI message is received that matches the VCT field.<br>This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Polarity (POL):** This specifies the polarity of each interrupt input.<br>When cleared, the signal is active high.<br>When set, the signal is active low.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO/V | **Delivery Status (DS):** This field contains the current status of the delivery of this interrupt.<br>When set, an interrupt is pending and not yet delivered.<br>When cleared, there is no activity for this entry<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode.<br>• These encodings are:<br>• 000 - Fixed<br>• 001 - Lowest Priority<br>• 010 - SMI (Not supported)<br>• 011 - Reserved<br>• 100 - NMI (Not supported)<br>• 101 - INIT (Not supported)<br>• 110 - Reserved<br>• 111 - ExtINT<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Vector (VCT):** This field contains the interrupt vector for this interrupt.<br>Values range between 10h and FEh.<br><br>**Power Well:** Core |

## 64.4.6 Redirection Table Entry 119 (RTE119)—Offset FEh

**Type:** IOxAPIC IDX/WDW (Size: 64 bits)

**IDX:** FCh, FDh

**Default:** 0000000000010000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 63:56 | 0x0<br>RW | **Destination ID (DID):** Destination ID of the local APIC<br><br>**Power Well:** Core |
| 55:48 | 0x0<br>RW | **Extended Destination ID (EDID):** Extended destination ID of the local APIC.<br><br>**Power Well:** Core |
| 47:17 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x1<br>RW | **Mask (MSK):**<br>When set, interrupts are not delivered nor held pending.<br>When cleared, and edge or level on this interrupt results in the delivery of the interrupt.<br><br>**Power Well:** Core |
| 15 | 0x0<br>RW | **Trigger Mode (TM):**<br>When cleared, the interrupt is edge sensitive.<br>When set, the interrupt is level sensitive.<br><br>**Power Well:** Core |
| 14 | 0x0<br>RO/V | **Remote IRR (RIRR):** This is used for level triggered interrupts - its meaning is undefined for edge triggered interrupts.<br>This bit is set when IOxAPIC sends the level interrupt message to the CPU.<br>This bit is cleared when an EOI message is received that matches the VCT field.<br>This bit is never set for SMI, NMI, INIT, or ExtINT delivery modes.<br><br>**Power Well:** Core |
| 13 | 0x0<br>RW | **Polarity (POL):** This specifies the polarity of each interrupt input.<br>When cleared, the signal is active high.<br>When set, the signal is active low.<br><br>**Power Well:** Core |
| 12 | 0x0<br>RO/V | **Delivery Status (DS):** This field contains the current status of the delivery of this interrupt.<br>When set, an interrupt is pending and not yet delivered.<br>When cleared, there is no activity for this entry<br><br>**Power Well:** Core |
| 11 | 0x0<br>RW | **Destination Mode (DSM):** This field is used by the local APIC to determine whether it is the destination of the message.<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10:8 | 0x0<br>RW | **Delivery Mode (DLM):** This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode.<br>These encodings are:<br>• 000 - Fixed<br>• 001 - Lowest Priority<br>• 010 - SMI (Not supported)<br>• 011 - Reserved<br>• 100 - NMI (Not supported)<br>• 101 - INIT (Not supported)<br>• 110 - Reserved<br>• 111 - ExtINT<br><br>**Power Well:** Core |
| 7:0 | 0x0<br>RW | **Vector (VCT):** This field contains the interrupt vector for this interrupt.<br>Values range between 10h and FEh.<br><br>**Power Well:** Core |

## 64.5 Registers in I/O Space—Fixed Addresses

These registers are accessed in Host I/O Space through fixed I/O Port addresses.

### 64.5.1 Master Initialization Command Word 1 (MICW1)—I/O Port 20h

A write to Initialization Command Word 1 (ICW1) starts the interrupt controller initialization sequence, during which the following occurs:

1. The Interrupt Mask register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.

Once this write occurs, the controller expects writes to ICW2, ICW3, and ICW4 to complete the initialization sequence.

**Type:** I/O Register (Size: 8 bits)

**I/O Port Address:** 20h

**Default:** 11h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **ICW/OCW select (ICW_OCW_SLT1):** These bits are MCS-85 specific, and not needed. Should be programmed to 000<br><br>**Power Well:** Core |
| 4 | 0x1<br>WO | **ICW/OCW select (ICW_OCW_SLT2):** This bit must be a 1 to select ICW1 and enable the ICW2, ICW3, and ICW4 sequence.<br><br>**Power Well:** Core |
| 3 | 0x0<br>WO | **Edge/Level Bank Select (LTIM):** Disabled. Replaced by the edge/level triggered control registers (ELCR).<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **ADI-IGNORED (ADI):** Ignored by the SoC. Should be programmed to 0.<br><br>**Power Well:** Core |
| 1 | 0x0<br>WO | **Single or Cascade (SNGL):** Must be programmed to a 0 to indicate two controllers operating in cascade mode.<br><br>**Power Well:** Core |
| 0 | 0x1<br>WO | **ICW4 Write Required (IC4):** This bit must be programmed to a 1 to indicate that ICW4 needs to be programmed.<br><br>**Power Well:** Core |

## 64.5.2 Master Initialization Command Word 2 (MICW2)—I/O Port 21h

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the CPU to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for the master controller and 70h for the slave controller.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 21h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **Interrupt Vector Base Address (IVBA):** Bits [7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>WO | **Interrupt Request Level (IRL):** When writing ICW2, these bits should all be 0. During an interrupt acknowledge cycle, these bits are programmed by the interrupt controller with the interrupt to be serviced. This is combined with bits [7:3] to form the interrupt vector driven onto the data bus during the second INTA# cycle.<br>The code is a three bit binary code:<br>Code Master Interrupt Slave Interrupt<br>• 000 IRQ0 IRQ8<br>• 001 IRQ1 IRQ9<br>• 010 IRQ2 IRQ10<br>• 011 IRQ3 IRQ11<br>• 100 IRQ4 IRQ12<br>• 101 IRQ5 IRQ13<br>• 110 IRQ6 IRQ14<br>• 111 IRQ7 IRQ15<br><br>**Power Well:** Core |

## 64.5.3 Master Initialization Command Word 3 (MICW3)—I/O Port 21h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 21h

**Default:** 07h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **MICW3 [7:3] (MICW3_7_3):** These bits must be programmed to zero.<br><br>**Power Well:** Core |
| 2 | 0x1<br>WO | **Cascaded Controller Connection (CCC):** This bit must always be programmed to a 1 to indicate the slave controller for interrupts 8 - 15 is cascaded on IRQ2.<br><br>**Power Well:** Core |
| 1:0 | 0x3<br>WO | **MICW [1:0] (MICW3_1_0):** These bits must be programmed to zero.<br><br>**Power Well:** Core |

## 64.5.4 Master Initialization Command Word 4 (MICW4)—I/O Port 21h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 21h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |
| 4 | 0x0<br>WO | **Special Fully Nested Mode (SFNM):** Should normally be disabled by writing a 0 to this bit.<br>If SFNM=1, the special fully nested mode is programmed.<br><br>**Power Well:** Core |
| 3 | 0x0<br>WO | **Buffered Mode (BUF):** Must be cleared for non-buffered mode.<br>Writing 1 will result in undefined behavior.<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **Master/Slave in Buffered Mode (MSBM):** Not used. Should always be programmed to 0.<br><br>**Power Well:** Core |
| 1 | 0x0<br>WO | **Automatic End of Interrupt (AEOI):** This bit should normally be programmed to 0.<br>This is the normal end of interrupt. If this bit is 1, the automatic end of interrupt mode is programmed. AEOI is discussed in Section 13.1.7.2<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Microprocessor Mode (MM):** This bit must be written to 1 to indicate that the controller is operating in an Intel Architecture-based system.<br>Writing 0 will result in undefined behavior.1<br><br>**Power Well:** Core |

## 64.5.5 Master Operational Control Word 1 (MOCW1)—I/O Port 21h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 21h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Request Mask (IRM):** When a 1 is written to any bit in this register, the corresponding IRQ line is masked.<br>When a 0 is written to any bit in this register, the corresponding IRQ mask bit is cleared, and interrupt requests will again be accepted by the controller.<br>Masking IRQ2 on the master controller will also mask the interrupt requests from the slave controller.<br><br>**Power Well:** Core |

## 64.5.6 Master Operational Control Word 2 (MOCW2)—I/O Port 20h

Following a part reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 20h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **Rotate and EOI Codes (REOI):** R, SL, EOI - These three bits control the Rotate and End of Interrupt modes and combinations of the two. A chart of these combinations is listed above under the bit definition.<br>• 000 Rotate in Auto EOI Mode (Clear)<br>• 001 Non-specific EOI command<br>• 010 No Operation<br>• 011 *Specific EOI Command<br>• 100 Rotate in Auto EOI Mode (Set)<br>• 101 Rotate on Non-Specific EOI Command<br>• 110 *Set Priority Command<br>• 111 *Rotate on Specific EOI Command<br>• *L0 - L2 Are Used<br><br>**Power Well:** Core |
| 4:3 | 0x0<br>WO | **OCW2 Select (O2S):** When selecting OCW2, bits 4:3 = 00<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>WO | **Interrupt Level Select (L2, L1, L0) (ILSLT):** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active.<br>A simple binary code, outlined above, selects the channel for the command to act upon. When the SL bit is inactive, these bits do not have a defined function, programming L2, L1 and L0 to 0 is sufficient in this case.<br><br>**Power Well:** Core |

## 64.5.7 Master Operational Control Word 3 (MOCW3)—I/O Port 20h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 20h

**Default:** 08h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |
| 6 | 0x0<br>WO | **Special Mask Mode (SMM):** If this bit is set, the Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/ disabling of the other channel's mask bits. Bit 6, the ESMM bit, must be set for this bit to have any meaning.<br><br>**Power Well:** Core |
| 5 | 0x0<br>WO | **Enable Special Mask Mode (ESMM):**<br><br>**Power Well:** Core |
| 4:3 | 0x1<br>WO | **OCW3 Select (O3S):** When selecting OCW3, bits 4:3 = 01<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **Poll Mode Command (PMC):** When cleared, poll command is not issued. When set, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle.<br>An encoded byte is driven onto the data bus, representing the highest priority level requesting service.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>WO | **Register Read Command (RRC):** . To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read.<br>Value Command<br>• 00 No Action<br>• 01 No Action<br>• 10 Read IRQ Register<br>• 11 Read IS Register<br><br>**Power Well:** Core |

## 64.5.8 Slave Initialization Command Word 1 (SICW1)—I/O Port A0h

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence, during which the following occurs: 1. The Interrupt Mask register is cleared. 2. IRQ7 input is assigned priority 7. 3. The slave mode address is set to 7. 4. Special Mask Mode is cleared and Status Read is set to IRR. Once this write occurs, the controller expects writes to ICW2, ICW3, and ICW4 to complete the initialization sequence.

**Type:** I/O Register (Size: 8 bits)

**I/O Port Address:** A0h

**Default:** 11h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **ICW/OCW select (ICW_OCW_SLT1):** These bits are MCS-85 specific, and not needed. Should be programmed to 000<br><br>**Power Well:** Core |
| 4 | 0x1<br>WO | **ICW/OCW select (ICW_OCW_SLT2):** This bit must be a 1 to select ICW1 and enable the ICW2, ICW3, and ICW4 sequence.<br><br>**Power Well:** Core |
| 3 | 0x0<br>WO | **Edge/Level Bank Select (LTIM):** Disabled. Replaced by the edge/level triggered control registers (ELCR).<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **ADI-IGNORED (ADI):** Ignored by the SoC. Should be programmed to 0.<br><br>**Power Well:** Core |
| 1 | 0x0<br>WO | **Single or Cascade (SNGL):** Must be programmed to a 0 to indicate two controllers operating in cascade mode.<br><br>**Power Well:** Core |
| 0 | 0x1<br>WO | **ICW4 Write Required (IC4):** This bit must be programmed to a 1 to indicate that ICW4 needs to be programmed.<br><br>**Power Well:** Core |

## 64.5.9 Slave Initialization Command Word 2 (SICW2)—I/O Port A1h

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the CPU to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for the master controller and 70h for the slave controller.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A1h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **Interrupt Vector Base Address (IVBA):** Bits [7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>WO | **Interrupt Request Level (IRL):** When writing ICW2, these bits should all be 0. During an interrupt acknowledge cycle, these bits are programmed by the interrupt controller with the interrupt to be serviced. This is combined with bits [7:3] to form the interrupt vector driven onto the data bus during the second INTA# cycle.<br>The code is a three bit binary code:<br>Code Master Interrupt Slave Interrupt<br>• 000 IRQ0 IRQ8<br>• 001 IRQ1 IRQ9<br>• 010 IRQ2 IRQ10<br>• 011 IRQ3 IRQ11<br>• 100 IRQ4 IRQ12<br>• 101 IRQ5 IRQ13<br>• 110 IRQ6 IRQ14<br>• 111 IRQ7 IRQ15<br><br>**Power Well:** Core |

## 64.5.10 Slave Initialization Command Word 3 (SICW3)—I/O Port A1h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A1h

**Default:** 07h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>WO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |
| 2:0 | 0x7<br>WO | **Slave Identification Code (SIC):** This field must be programmed to 02h to match the code broadcast by the master controller during the INTA# sequence.<br><br>**Power Well:** Core |

## 64.5.11 Slave Initialization Command Word 4 (SICW4)—I/O Port A1h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A1h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |
| 4 | 0x0<br>WO | **Special Fully Nested Mode (SFNM):** Should normally be disabled by writing a 0 to this bit.<br>If SFNM=1, the special fully nested mode is programmed.<br><br>**Power Well:** Core |
| 3 | 0x0<br>WO | **Buffered Mode (BUF):** Must be cleared for non-buffered mode. Writing 1 will result in undefined behavior.<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **Master/Slave in Buffered Mode (MSBM):** Not used. Should always be programmed to 0.<br><br>**Power Well:** Core |
| 1 | 0x0<br>WO | **Automatic End of Interrupt (AEOI):** This bit should normally be programmed to 0.<br>This is the normal end of interrupt.<br>If this bit is 1, the automatic end of interrupt mode is programmed. AEOI is discussed in Section 13.1.7.2<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **Microprocessor Mode (MM):** This bit must be written to 1 to indicate that the controller is operating in an Intel Architecture-based system.<br>Writing 0 will result in undefined behavior.1<br><br>**Power Well:** Core |

## 64.5.12 Slave Operational Control Word 1 (SOCW1)—I/O Port A1h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A1h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:0 | 0x0<br>RW | **Interrupt Request Mask (IRM):** When a 1 is written to any bit in this register, the corresponding IRQ line is masked. When a 0 is written to any bit in this register, the corresponding IRQ mask bit is cleared, and interrupt requests will again be accepted by the controller.<br>Masking IRQ2 on the master controller will also mask the interrupt requests from the slave controller.<br><br>**Power Well:** Core |

## 64.5.13 Slave Operational Control Word 2 (SOCW2)—I/O Port A0h

Following a part reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:5 | 0x0<br>WO | **Rotate and EOI Codes (REOI):** R, SL, EOI - These three bits control the Rotate and End of Interrupt modes and combinations of the two.<br>A chart of these combinations is listed above under the bit definition.<br>• 000 Rotate in Auto EOI Mode (Clear)<br>• 001 Non-specific EOI command<br>• 010 No Operation<br>• 011 *Specific EOI Command<br>• 100 Rotate in Auto EOI Mode (Set)<br>• 101 Rotate on Non-Specific EOI Command<br>• 110 *Set Priority Command<br>• 111 *Rotate on Specific EOI Command<br>*L0 - L2 Are Used<br><br>**Power Well:** Core |
| 4:3 | 0x0<br>WO | **OCW2 Select (O2S):** When selecting OCW2, bits 4:3 = 00<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>WO | **Interrupt Level Select (L2, L1, L0) (ILSLT):** L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active.<br>A simple binary code, outlined above, selects the channel for the command to act upon.<br>When the SL bit is inactive, these bits do not have a defined function, programming L2, L1 and L0 to 0 is sufficient in this case.<br><br>**Power Well:** Core |

## 64.5.14 Slave Operational Control Word 3 (SOCW3)—I/O Port A0h

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** A0h

**Default:** 08h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RO | **Reserved (RSVD):** Must be 0.<br><br>**Power Well:** Core |
| 6 | 0x0<br>WO | **Special Mask Mode (SMM):** If this bit is set, the Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/ disabling of the other channel's mask bits.<br>Bit 6, the ESMM bit, must be set for this bit to have any meaning.<br><br>**Power Well:** Core |
| 5 | 0x0<br>WO | **Enable Special Mask Mode (ESMM):**<br><br>**Power Well:** Core |
| 4:3 | 0x1<br>WO | **OCW3 Select (O3S):** When selecting OCW3, bits 4:3 = 01<br><br>**Power Well:** Core |
| 2 | 0x0<br>WO | **Poll Mode Command (PMC):** When cleared, poll command is not issued.<br>When set, the next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle.<br>An encoded byte is driven onto the data bus, representing the highest priority level requesting service.<br><br>**Power Well:** Core |
| 1:0 | 0x0<br>WO | **Register Read Command (RRC):** To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read.<br>Value Command<br>• 00 No Action<br>• 01 No Action<br>• 10 Read IRQ Register<br>• 11 Read IS Register<br><br>**Power Well:** Core |

## 64.5.15 Master Edge/Level Control (ELCR1)—I/O Port 4D0h

Master Edge/Level Control Register

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 4D0h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:3 | 0x0<br>RW | **Edge Level Control (ELC_7_3):** In edge mode, (bit cleared), the interrupt is recognized by a low to high transition.<br>In level mode (bit set), the interrupt is recognized by a high level.<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RO | **Reserved (RSVD):** The cascade channel(IRQ2), heart beat timer (IRQ0), and keyboard controller (IRQ1), cannot be put into level mode<br><br>**Power Well:** Core |

## 64.5.16 Slave Edge/Level Control (ELCR2)—I/O Port 4D1h

Slave Edge/Level Control Register

**Type:** I/O Register
(Size: 8 bits)

**I/O Port Address:** 4D1h

**Default:** 00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7:6 | 0x0<br>RW | **Edge Level Control (ELC_15_14):** In edge mode, (bit cleared), the interrupt is recognized by a low to high transition.<br>In level mode (bit set), the interrupt is recognized by a high level. Bit 7 applies to IRQ15, and bit 6 to IRQ14.<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW | **Edge Level Control (ELC_13):** In edge mode, (bit cleared), the interrupt is recognized by a low to high transition.<br>In level mode (bit set), the interrupt is recognized by a high level.<br>This bit applies to IRQ13.<br><br>**Power Well:** Core |
| 4:1 | 0x0<br>RW | **Edge Level Control (ELC_12_9):** In edge mode, (bit cleared), the interrupt is recognized by a low to high transition.<br>In level mode (bit set), the interrupt is recognized by a high level. Bit 4 applies to IRQ12, bit 3 to IRQ11, bit 2 to IRQ10, and bit 1 to IRQ9.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RO | **Reserved (RSVD):** The Real Time Clock (IRQ8#) cannot be programmed for level mode.<br><br>**Power Well:** Core |

§ §

# 65 Customer General-Purpose I/O Controller

## 65.1 Introduction and Index

The host-accessible registers for the Customer General-Purpose I/O Controller are described here. Also, the registers that allow the selection of one of a number of shared, signal functions are also described here.

Besides the Host, the customer general-purpose I/O controller are also accessible through Intel® ME Configuration Space and IE configuration Space using their own Bus, Device and Function Numbers.

The legend for the register access codes are described in Chapter 43, "Introduction."

*Note:* The following statement needs to be noted when referring to register Table 65-1 and Table 65-2:

The internal pull-up resistor option is not supported for 1.8V GPIOs.
The internal pull-down resistor option is not supported for 3.3V GPIOs.

### 65.1.1 Sideband Registers—Configurable I/O Signals

An index of the Sideband Registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method."

**Table 65-1. Summary of Sideband Registers, Part 1 of 2—Sideband Port 0xC2 (Sheet 1 of 4)**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| C | 00000400 | "Pad Base Address (PADBAR)—Offset Ch" on page 2953 |
| 10 | 00021000 | "Miscellaneous Configuration (MISCCFG)—Offset 10h" on page 2954 |
| 20 | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_0)—Offset 20h" on page 2955 |
| 24 | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_1)—Offset 24h" on page 2957 |
| 28 | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_2)—Offset 28h" on page 2959 |
| 2C | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_3)—Offset 2Ch" on page 2961 |
| 30 | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_4)—Offset 30h" on page 2963 |
| 34 | 00000000 | "Pad Ownership (PAD_OWN_NORTH_ALL_5)—Offset 34h" on page 2965 |
| 90 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_0)—Offset 90h" on page 2966 |
| 94 | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_0)—Offset 94h" on page 2968 |
| 98 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_1)—Offset 98h" on page 2970 |
| 9C | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_1)—Offset 9Ch" on page 2971 |
| C0 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_0)—Offset C0h" on page 2972 |
| C4 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_1)—Offset C4h" on page 2974 |
| 100 | 00000000 | "GPI Interrupt Status (GPI_IS_NORTH_ALL_0)—Offset 100h" on page 2975 |
| 104 | 00000000 | "GPI Interrupt Status (GPI_IS_NORTH_ALL_1)—Offset 104h" on page 2977 |
| 120 | 00000000 | "GPI Interrupt Enable (GPI_IE_NORTH_ALL_0)—Offset 120h" on page 2978 |
| 124 | 00000000 | "GPI Interrupt Enable (GPI_IE_NORTH_ALL_1)—Offset 124h" on page 2980 |
| 140 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_0)—Offset 140h" on page 2981 |
| 144 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_1)—Offset 144h" on page 2983 |
| 160 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_0)—Offset 160h" on page 2984 |
| 164 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_1)—Offset 164h" on page 2986 |
| 180 | 00000000 | "SMI Status (GPI_SMI_STS_NORTH_ALL_0)—Offset 180h" on page 2987 |
| 184 | 00000000 | "SMI Status (GPI_SMI_STS_NORTH_ALL_1)—Offset 184h" on page 2989 |
| 1A0 | 00000000 | "SMI Enable (GPI_SMI_EN_NORTH_ALL_0)—Offset 1A0h" on page 2990 |
| 1A4 | 00000000 | "SMI Enable (GPI_SMI_EN_NORTH_ALL_1)—Offset 1A4h" on page 2992 |
| 1C0 | 00000000 | "NMI Status (GPI_NMI_STS_NORTH_ALL_0)—Offset 1C0h" on page 2993 |
| 1C4 | 00000000 | "NMI Status (GPI_NMI_STS_NORTH_ALL_1)—Offset 1C4h" on page 2995 |
| 1E0 | 00000000 | "NMI Enable (GPI_NMI_EN_NORTH_ALL_0)—Offset 1E0h" on page 2996 |
| 1E4 | 00000000 | "NMI Enable (GPI_NMI_EN_NORTH_ALL_1)—Offset 1E4h" on page 2998 |
| 400 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP0)—Offset 400h" on page 2999 |
| 404 | 00000018 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP0)—Offset 404h" on page 3002 |

## Table 65-1. Summary of Sideband Registers, Part 1 of 2—Sideband Port 0xC2 (Sheet 2 of 4)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 408 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP0)—Offset 408h" on page 3003 |
| 40C | 00000019 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP0)—Offset 40Ch" on page 3006 |
| 410 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP1)—Offset 410h" on page 3007 |
| 414 | 0000001A | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP1)—Offset 414h" on page 3010 |
| 418 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP1)—Offset 418h" on page 3011 |
| 41C | 0000001B | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP1)—Offset 41Ch" on page 3014 |
| 420 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP2)—Offset 420h" on page 3015 |
| 424 | 0000001C | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP2)—Offset 424h" on page 3018 |
| 428 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP2)—Offset 428h" on page 3019 |
| 42C | 0000001D | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP2)—Offset 42Ch" on page 3022 |
| 430 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP3)—Offset 430h" on page 3023 |
| 434 | 0000001E | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP3)—Offset 434h" on page 3026 |
| 438 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP3)—Offset 438h" on page 3027 |
| 43C | 0000001F | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP3)—Offset 43Ch" on page 3030 |
| 440 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED0)—Offset 440h" on page 3031 |
| 444 | 00000020 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED0)—Offset 444h" on page 3034 |
| 448 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED1)—Offset 448h" on page 3035 |
| 44C | 00000021 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED1)—Offset 44Ch" on page 3038 |
| 450 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_CLK)—Offset 450h" on page 3039 |
| 454 | 00002822 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_CLK)—Offset 454h" on page 3042 |
| 458 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_DATA)—Offset 458h" on page 3043 |
| 45C | 00002823 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_DATA)—Offset 45Ch" on page 3046 |
| 460 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_CLK)—Offset 460h" on page 3047 |
| 464 | 00002824 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_CLK)—Offset 464h" on page 3050 |
| 468 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_DATA)—Offset 468h" on page 3051 |
| 46C | 00002825 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_DATA)—Offset 46Ch" on page 3054 |
| 470 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD0)—Offset 470h" on page 3055 |
| 474 | 00003026 | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD0)—Offset 474h" on page 3058 |
| 478 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CLK_IN)—Offset 478h" on page 3059 |
| 47C | 00003027 | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CLK_IN)—Offset 47Ch" on page 3062 |
| 480 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD1)—Offset 480h" on page 3063 |
| 484 | 00003028 | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD1)—Offset 484h" on page 3066 |
| 488 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CRS_DV)—Offset 488h" on page 3067 |
| 48C | 00000029 | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CRS_DV)—Offset 48Ch" on page 3070 |
| 490 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_IN)—Offset 490h" on page 3071 |
| 494 | 0000102A | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_IN)—Offset 494h" on page 3074 |
| 498 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TX_EN)—Offset 498h" on page 3075 |
| 49C | 0000002B | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TX_EN)—Offset 49Ch" on page 3078 |
| 4A0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD0)—Offset 4A0h" on page 3079 |

## Table 65-1. Summary of Sideband Registers, Part 1 of 2—Sideband Port 0xC2 (Sheet 3 of 4)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 4A4 | 0000002C | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD0)—Offset 4A4h" on page 3082 |
| 4A8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD1)—Offset 4A8h" on page 3083 |
| 4AC | 0000002D | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD1)—Offset 4ACh" on page 3086 |
| 4B0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_OUT)—Offset 4B0h" on page 3087 |
| 4B4 | 0000002E | "Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_OUT)—Offset 4B4h" on page 3090 |
| 4B8 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED0)—Offset 4B8h" on page 3091 |
| 4BC | 0000002F | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED0)—Offset 4BCh" on page 3094 |
| 4C0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED1)—Offset 4C0h" on page 3095 |
| 4C4 | 00000030 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED1)—Offset 4C4h" on page 3098 |
| 4C8 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED0)—Offset 4C8h" on page 3099 |
| 4CC | 00000031 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED0)—Offset 4CCh" on page 3102 |
| 4D0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED1)—Offset 4D0h" on page 3103 |
| 4D4 | 00000032 | "Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED1)—Offset 4D4h" on page 3106 |
| 4D8 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_0)—Offset 4D8h" on page 3107 |
| 4DC | 00000033 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_0)—Offset 4DCh" on page 3110 |
| 4E0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ0_N)—Offset 4E0h" on page 3111 |
| 4E4 | 00000034 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ0_N)—Offset 4E4h" on page 3114 |
| 4E8 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ1_N)—Offset 4E8h" on page 3115 |
| 4EC | 00000035 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ1_N)—Offset 4ECh" on page 3118 |
| 4F0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ2_N)—Offset 4F0h" on page 3119 |
| 4F4 | 00000036 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ2_N)—Offset 4F4h" on page 3122 |
| 4F8 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ3_N)—Offset 4F8h" on page 3123 |
| 4FC | 00000037 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ3_N)—Offset 4FCh" on page 3126 |
| 500 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ4_N)—Offset 500h" on page 3127 |
| 504 | 00000038 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ4_N)—Offset 504h" on page 3130 |
| 508 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_1)—Offset 508h" on page 3131 |
| 50C | 00000039 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_1)—Offset 50Ch" on page 3134 |
| 510 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_2)—Offset 510h" on page 3135 |
| 514 | 0000003A | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_2)—Offset 514h" on page 3138 |
| 518 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SVID_ALERT_N)—Offset 518h" on page 3139 |
| 51C | 0000003B | "Pad Configuration DW1 (PAD_CFG_DW1_SVID_ALERT_N)—Offset 51Ch" on page 3142 |
| 520 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SVID_DATA)—Offset 520h" on page 3143 |
| 524 | 0000003C | "Pad Configuration DW1 (PAD_CFG_DW1_SVID_DATA)—Offset 524h" on page 3146 |
| 528 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SVID_CLK)—Offset 528h" on page 3147 |
| 52C | 0000003D | "Pad Configuration DW1 (PAD_CFG_DW1_SVID_CLK)—Offset 52Ch" on page 3150 |
| 530 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_THERMTRIP_N)—Offset 530h" on page 3151 |
| 534 | 0000303E | "Pad Configuration DW1 (PAD_CFG_DW1_THERMTRIP_N)—Offset 534h" on page 3154 |
| 538 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PROCHOT_N)—Offset 538h" on page 3155 |

**Table 65-1. Summary of Sideband Registers, Part 1 of 2—Sideband Port 0xC2 (Sheet 4 of 4)**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 53C | 0000303F | "Pad Configuration DW1 (PAD_CFG_DW1_PROCHOT_N)—Offset 53Ch" on page 3158 |
| 540 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_MEMHOT_N)—Offset 540h" on page 3159 |
| 544 | 00003040 | "Pad Configuration DW1 (PAD_CFG_DW1_MEMHOT_N)—Offset 544h" on page 3162 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 1 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| C | 00000400 | "Pad Base Address (PADBAR)—Offset 0Ch" on page 3163 |
| 10 | 00021000 | "Miscellaneous Configuration (MISCCFG)—Offset 10h" on page 3164 |
| 20 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_DFX_0)—Offset 20h" on page 3165 |
| 24 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_DFX_1)—Offset 24h" on page 3167 |
| 28 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_DFX_2)—Offset 28h" on page 3169 |
| 2C | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_0)—Offset 2Ch" on page 3170 |
| 30 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_1)—Offset 30h" on page 3172 |
| 34 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_2)—Offset 34h" on page 3174 |
| 38 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_3)—Offset 38h" on page 3176 |
| 3C | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_4)—Offset 3Ch" on page 3178 |
| 40 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_5)—Offset 40h" on page 3180 |
| 44 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP0_6)—Offset 44h" on page 3182 |
| 48 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_0)—Offset 48h" on page 3183 |
| 4C | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_1)—Offset 4Ch" on page 3185 |
| 50 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_2)—Offset 50h" on page 3187 |
| 54 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_3)—Offset 54h" on page 3189 |
| 58 | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_4)—Offset 58h" on page 3191 |
| 5C | 00000000 | "Pad Ownership (PAD_OWN_SOUTH_GROUP1_5)—Offset 5Ch" on page 3193 |
| 90 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_SOUTH_DFX_0)—Offset 90h" on page 3194 |
| 94 | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_SOUTH_DFX_0)—Offset 94h" on page 3196 |
| 98 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_0)—Offset 98h" on page 3198 |
| 9C | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_0)—Offset 9Ch" on page 3200 |
| A0 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_1)—Offset A0h" on page 3202 |
| A4 | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_1)—Offset A4h" on page 3204 |
| A8 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_0)—Offset A8h" on page 3206 |
| AC | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_0)—Offset ACh" on page 3208 |
| B0 | 00000000 | "Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_1)—Offset B0h" on page 3210 |
| B4 | 00000000 | "Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_1)—Offset B4h" on page 3211 |
| C0 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_SOUTH_DFX_0)—Offset C0h" on page 3212 |
| C4 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_0)—Offset C4h" on page 3214 |
| C8 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_1)—Offset C8h" on page 3216 |
| CC | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_0)—Offset CCh" on page 3218 |
| D0 | 00000000 | "Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_1)—Offset D0h" on page 3220 |
| 100 | 00000000 | "GPI Interrupt Status (GPI_IS_SOUTH_DFX_0)—Offset 100h" on page 3221 |
| 104 | 00000000 | "GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_0)—Offset 104h" on page 3223 |
| 108 | 00000000 | "GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_1)—Offset 108h" on page 3225 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 2 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 10C | 00000000 | "GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_0)—Offset 10Ch" on page 3227 |
| 110 | 00000000 | "GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_1)—Offset 110h" on page 3229 |
| 120 | 00000000 | "GPI Interrupt Enable (GPI_IE_SOUTH_DFX_0)—Offset 120h" on page 3230 |
| 124 | 00000000 | "GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_0)—Offset 124h" on page 3232 |
| 128 | 00000000 | "GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_1)—Offset 128h" on page 3234 |
| 12C | 00000000 | "GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_0)—Offset 12Ch" on page 3236 |
| 130 | 00000000 | "GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_1)—Offset 130h" on page 3238 |
| 140 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_DFX_0)—Offset 140h" on page 3239 |
| 144 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_0)—Offset 144h" on page 3241 |
| 148 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_1)—Offset 148h" on page 3243 |
| 14C | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_0)—Offset 14Ch" on page 3245 |
| 150 | 00000000 | "GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_1)—Offset 150h" on page 3247 |
| 160 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_DFX_0)—Offset 160h" on page 3248 |
| 164 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_0)—Offset 164h" on page 3250 |
| 168 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_1)—Offset 168h" on page 3252 |
| 16C | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_0)—Offset 16Ch" on page 3254 |
| 170 | 00000000 | "GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_1)—Offset 170h" on page 3256 |
| 184 | 00000000 | "SMI Status (GPI_SMI_STS_SOUTH_GROUP0_0)—Offset 184h" on page 3257 |
| 188 | 00000000 | "SMI Status (GPI_SMI_STS_SOUTH_GROUP0_1)—Offset 188h" on page 3259 |
| 18C | 00000000 | "SMI Status (GPI_SMI_STS_SOUTH_GROUP1_0)—Offset 18Ch" on page 3261 |
| 190 | 00000000 | "SMI Status (GPI_SMI_STS_SOUTH_GROUP1_1)—Offset 190h" on page 3263 |
| 1A4 | 00000000 | "SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_0)—Offset 1A4h" on page 3264 |
| 1A8 | 00000000 | "SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_1)—Offset 1A8h" on page 3266 |
| 1AC | 00000000 | "SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_0)—Offset 1ACh" on page 3268 |
| 1B0 | 00000000 | "SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_1)—Offset 1B0h" on page 3270 |
| 1C4 | 00000000 | "NMI Status (GPI_NMI_STS_SOUTH_GROUP0_0)—Offset 1C4h" on page 3271 |
| 1C8 | 00000000 | "NMI Status (GPI_NMI_STS_SOUTH_GROUP0_1)—Offset 1C8h" on page 3273 |
| 1CC | 00000000 | "NMI Status (GPI_NMI_STS_SOUTH_GROUP1_0)—Offset 1CCh" on page 3275 |
| 1D0 | 00000000 | "NMI Status (GPI_NMI_STS_SOUTH_GROUP1_1)—Offset 1D0h" on page 3277 |
| 1E4 | 00000000 | "NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_0)—Offset 1E4h" on page 3278 |
| 1E8 | 00000000 | "NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_1)—Offset 1E8h" on page 3280 |
| 1EC | 00000000 | "NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_0)—Offset 1ECh" on page 3282 |
| 1F0 | 00000000 | "NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_1)—Offset 1F0h" on page 3284 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 3 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 400 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK0)—Offset 400h" on page 3285 |
| 404 | 00000041 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK0)—Offset 404h" on page 3288 |
| 408 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK1)—Offset 408h" on page 3289 |
| 40C | 00000042 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK1)—Offset 40Ch" on page 3292 |
| 410 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT0)—Offset 410h" on page 3293 |
| 414 | 00000043 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT0)—Offset 414h" on page 3296 |
| 418 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT1)—Offset 418h" on page 3297 |
| 41C | 00000044 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT1)—Offset 41Ch" on page 3300 |
| 420 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT2)—Offset 420h" on page 3301 |
| 424 | 00000045 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT2)—Offset 424h" on page 3304 |
| 428 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT3)—Offset 428h" on page 3305 |
| 42C | 00000046 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT3)—Offset 42Ch" on page 3308 |
| 430 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT4)—Offset 430h" on page 3309 |
| 434 | 00000047 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT4)—Offset 434h" on page 3312 |
| 438 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT5)—Offset 438h" on page 3313 |
| 43C | 00000048 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT5)—Offset 43Ch" on page 3316 |
| 440 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT6)—Offset 440h" on page 3317 |
| 444 | 00000049 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT6)—Offset 444h" on page 3320 |
| 448 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT7)—Offset 448h" on page 3321 |
| 44C | 0000004A | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT7)—Offset 44Ch" on page 3324 |
| 450 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT8)—Offset 450h" on page 3325 |
| 454 | 0000004B | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT8)—Offset 454h" on page 3328 |
| 458 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT9)—Offset 458h" on page 3329 |
| 45C | 0000004C | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT9)—Offset 45Ch" on page 3332 |
| 460 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT10)—Offset 460h" on page 3333 |
| 464 | 0000004D | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT10)—Offset 464h" on page 3336 |
| 468 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT11)—Offset 468h" on page 3337 |
| 46C | 0000004E | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT11)—Offset 46Ch" on page 3340 |
| 470 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT12)—Offset 470h" on page 3341 |
| 474 | 0000004F | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT12)—Offset 474h" on page 3344 |
| 478 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT13)—Offset 478h" on page 3345 |
| 47C | 00000050 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT13)—Offset 47Ch" on page 3348 |
| 480 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT14)—Offset 480h" on page 3349 |
| 484 | 00000051 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT14)—Offset 484h" on page 3352 |
| 488 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT15)—Offset 488h" on page 3353 |
| 48C | 00000052 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT15)—Offset 48Ch" on page 3356 |
| 490 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_12)—Offset 490h" on page 3357 |
| 494 | 00000053 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_12)—Offset 494h" on page 3360 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 4 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 498 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_ALRT_N)—Offset 498h" on page 3361 |
| 49C | 00000054 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_ALRT_N)—Offset 49Ch" on page 3364 |
| 4A0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ5_N)—Offset 4A0h" on page 3365 |
| 4A4 | 00000055 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ5_N)—Offset 4A4h" on page 3368 |
| 4A8 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ6_N)—Offset 4A8h" on page 3369 |
| 4AC | 00000056 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ6_N)—Offset 4ACh" on page 3372 |
| 4B0 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ7_N)—Offset 4B0h" on page 3373 |
| 4B4 | 00000057 | "Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ7_N)—Offset 4B4h" on page 3376 |
| 4B8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_UART0_RXD)—Offset 4B8h" on page 3377 |
| 4BC | 00003058 | "Pad Configuration DW1 (PAD_CFG_DW1_UART0_RXD)—Offset 4BCh" on page 3380 |
| 4C0 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_UART0_TXD)—Offset 4C0h" on page 3381 |
| 4C4 | 00000059 | "Pad Configuration DW1 (PAD_CFG_DW1_UART0_TXD)—Offset 4C4h" on page 3384 |
| 4C8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_CLK)—Offset 4C8h" on page 3385 |
| 4CC | 0000305A | "Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_CLK)—Offset 4CCh" on page 3388 |
| 4D0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_DATA)—Offset 4D0h" on page 3389 |
| 4D4 | 0000305B | "Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_DATA)—Offset 4D4h" on page 3392 |
| 4D8 | 45000601 | "Pad Configuration DW0 (PAD_CFG_DW0_ERROR2_N)—Offset 4D8h" on page 3393 |
| 4DC | 0000005C | "Pad Configuration DW1 (PAD_CFG_DW1_ERROR2_N)—Offset 4DCh" on page 3396 |
| 4E0 | 45000601 | "Pad Configuration DW0 (PAD_CFG_DW0_ERROR1_N)—Offset 4E0h" on page 3397 |
| 4E4 | 0000005D | "Pad Configuration DW1 (PAD_CFG_DW1_ERROR1_N)—Offset 4E4h" on page 3400 |
| 4E8 | 45000601 | "Pad Configuration DW0 (PAD_CFG_DW0_ERROR0_N)—Offset 4E8h" on page 3401 |
| 4EC | 0000005E | "Pad Configuration DW1 (PAD_CFG_DW1_ERROR0_N)—Offset 4ECh" on page 3404 |
| 4F0 | 44000601 | "Pad Configuration DW0 (PAD_CFG_DW0_IERR_N)—Offset 4F0h" on page 3405 |
| 4F4 | 0000005F | "Pad Configuration DW1 (PAD_CFG_DW1_IERR_N)—Offset 4F4h" on page 3408 |
| 4F8 | 44000601 | "Pad Configuration DW0 (PAD_CFG_DW0_MCERR_N)—Offset 4F8h" on page 3409 |
| 4FC | 00000060 | "Pad Configuration DW1 (PAD_CFG_DW1_MCERR_N)—Offset 4FCh" on page 3412 |
| 500 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_CLK)—Offset 500h" on page 3413 |
| 504 | 00003061 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_CLK)—Offset 504h" on page 3416 |
| 508 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_DATA)—Offset 508h" on page 3417 |
| 50C | 00003062 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_DATA)—Offset 50Ch" on page 3420 |
| 510 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_ALRT_N)—Offset 510h" on page 3421 |
| 514 | 00003063 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_ALRT_N)—Offset 514h" on page 3424 |
| 518 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_DATA)—Offset 518h" on page 3425 |
| 51C | 00003064 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_DATA)—Offset 51Ch" on page 3428 |
| 520 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_CLK)—Offset 520h" on page 3429 |
| 524 | 00003065 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_CLK)—Offset 524h" on page 3432 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 5 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 528 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_DATA)—Offset 528h" on page 3433 |
| 52C | 00003066 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_DATA)—Offset 52Ch" on page 3436 |
| 530 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_CLK)—Offset 530h" on page 3437 |
| 534 | 00003067 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_CLK)—Offset 534h" on page 3440 |
| 538 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_DATA)—Offset 538h" on page 3441 |
| 53C | 00003068 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_DATA)—Offset 53Ch" on page 3444 |
| 540 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_CLK)—Offset 540h" on page 3445 |
| 544 | 00003069 | "Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_CLK)—Offset 544h" on page 3448 |
| 548 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_ALRT_N)—Offset 548h" on page 3449 |
| 54C | 0000306A | "Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_ALRT_N)—Offset 54Ch" on page 3452 |
| 550 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_USB_OC0_N)—Offset 550h" on page 3453 |
| 554 | 0000306B | "Pad Configuration DW1 (PAD_CFG_DW1_USB_OC0_N)—Offset 554h" on page 3456 |
| 558 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE0)—Offset 558h" on page 3457 |
| 55C | 0000006C | "Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE0)—Offset 55Ch" on page 3460 |
| 560 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE1)—Offset 560h" on page 3461 |
| 564 | 0000006D | "Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE1)—Offset 564h" on page 3464 |
| 568 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_4)—Offset 568h" on page 3465 |
| 56C | 0000006E | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_4)—Offset 56Ch" on page 3468 |
| 570 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_5)—Offset 570h" on page 3469 |
| 574 | 0000006F | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_5)—Offset 574h" on page 3472 |
| 578 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_6)—Offset 578h" on page 3473 |
| 57C | 00000070 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_6)—Offset 57Ch" on page 3476 |
| 580 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_7)—Offset 580h" on page 3477 |
| 584 | 00000071 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_7)—Offset 584h" on page 3480 |
| 588 | 45000700 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA0_LED_N)—Offset 588h" on page 3481 |
| 58C | 00000072 | "Pad Configuration DW1 (PAD_CFG_DW1_SATA0_LED_N)—Offset 58Ch" on page 3484 |
| 590 | 45000700 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA1_LED_N)—Offset 590h" on page 3485 |
| 594 | 00000073 | "Pad Configuration DW1 (PAD_CFG_DW1_SATA1_LED_N)—Offset 594h" on page 3488 |
| 598 | 45000700 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT0)—Offset 598h" on page 3489 |
| 59C | 00000074 | "Pad Configuration DW1 (PAD_CFG_DW1_SATA_PDETECT0)—Offset 59Ch" on page 3492 |
| 5A0 | 45000700 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT1)—Offset 5A0h" on page 3493 |
| 5A4 | 00000075 | "Pad Configuration DW1 (PAD_CFG_DW1)—Offset 5A4h" on page 3496 |
| 5A8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA0_SDOUT)—Offset 5A8h" on page 3497 |
| 5AC | 00000076 | "Pad Configuration DW1 (PAD_CFG_DW1_SATA0_SDOUT)—Offset 5ACh" on page 3500 |
| 5B0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_SATA1_SDOUT)—Offset 5B0h" on page 3501 |
| 5B4 | 00000018 | "Pad Configuration DW1 (PAD_CFG_DW1_SATA1_SDOUT)—Offset 5B4h" on page 3504 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 6 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 5B8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_UART1_RXD)—Offset 5B8h" on page 3505 |
| 5BC | 00003019 | "Pad Configuration DW1 (PAD_CFG_DW1_UART1_RXD)—Offset 5BCh" on page 3508 |
| 5C0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_UART1_TXD)—Offset 5C0h" on page 3509 |
| 5C4 | 0000001A | "Pad Configuration DW1 (PAD_CFG_DW1_UART1_TXD)—Offset 5C4h" on page 3512 |
| 5C8 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_8)—Offset 5C8h" on page 3513 |
| 5CC | 0000001B | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_8)—Offset 5CCh" on page 3516 |
| 5D0 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_9)—Offset 5D0h" on page 3517 |
| 5D4 | 0000001C | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_9)—Offset 5D4h" on page 3520 |
| 5D8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_TCK)—Offset 5D8h" on page 3521 |
| 5DC | 0400101D | "Pad Configuration DW1 (PAD_CFG_DW1_TCK)—Offset 5DCh" on page 3524 |
| 5E0 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_TRST_N)—Offset 5E0h" on page 3525 |
| 5E4 | 0400301E | "Pad Configuration DW1 (PAD_CFG_DW1_TRST_N)—Offset 5E4h" on page 3528 |
| 5E8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_TMS)—Offset 5E8h" on page 3529 |
| 5EC | 0400301F | "Pad Configuration DW1 (PAD_CFG_DW1_TMS)—Offset 5ECh" on page 3532 |
| 5F0 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_TDI)—Offset 5F0h" on page 3533 |
| 5F4 | 04003020 | "Pad Configuration DW1 (PAD_CFG_DW1_TDI)—Offset 5F4h" on page 3536 |
| 5F8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_TDO)—Offset 5F8h" on page 3537 |
| 5FC | 00000021 | "Pad Configuration DW1 (PAD_CFG_DW1_TDO)—Offset 5FCh" on page 3540 |
| 600 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_CX_PRDY_N)—Offset 600h" on page 3541 |
| 604 | 00003022 | "Pad Configuration DW1 (PAD_CFG_DW1_CX_PRDY_N)—Offset 604h" on page 3544 |
| 608 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_CX_PREQ_N)—Offset 608h" on page 3545 |
| 60C | 04003023 | "Pad Configuration DW1 (PAD_CFG_DW1_CX_PREQ_N)—Offset 60Ch" on page 3548 |
| 610 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGINOUT)—Offset 610h" on page 3549 |
| 614 | 04003024 | "Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGINOUT)—Offset 614h" on page 3552 |
| 618 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGOUT)—Offset 618h" on page 3553 |
| 61C | 04000025 | "Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGOUT)—Offset 61Ch" on page 3556 |
| 620 | 44000300 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE2)—Offset 620h" on page 3557 |
| 624 | 04000026 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE2)—Offset 624h" on page 3560 |
| 628 | 44000300 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE3)—Offset 628h" on page 3561 |
| 62C | 04000027 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE3)—Offset 62Ch" on page 3564 |
| 630 | 44000300 | "Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE4)—Offset 630h" on page 3565 |
| 634 | 00000028 | "Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE4)—Offset 634h" on page 3568 |
| 638 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SUSPWRDNACK)—Offset 638h" on page 3569 |
| 63C | 00000029 | "Pad Configuration DW1 (PAD_CFG_DW1_SUSPWRDNACK)—Offset 63Ch" on page 3572 |
| 640 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_SUSCLK)—Offset 640h" on page 3573 |
| 644 | 0000002A | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_SUSCLK)—Offset 644h" on page 3576 |
| 648 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_ADR_TRIGGER)—Offset 648h" on page 3577 |
| 64C | 0000002B | "Pad Configuration DW1 (PAD_CFG_DW1_ADR_TRIGGER)—Offset 64Ch" on page 3580 |
| 650 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S45_N)—Offset 650h" on page 3581 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 7 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 654 | 0000002C | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S45_N)—Offset 654h" on page 3584 |
| 658 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S3_N)—Offset 658h" on page 3585 |
| 65C | 0000002D | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S3_N)—Offset 65Ch" on page 3588 |
| 660 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_WAKE_N)—Offset 660h" on page 3589 |
| 664 | 0000302E | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_WAKE_N)—Offset 664h" on page 3592 |
| 668 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_PWRBTN_N)—Offset 668h" on page 3593 |
| 66C | 0000302F | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_PWRBTN_N)—Offset 66Ch" on page 3596 |
| 670 | 45000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_RESETBUTTON_N)—Offset 670h" on page 3597 |
| 674 | 00003030 | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_RESETBUTTON_N)—Offset 674h" on page 3600 |
| 678 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_PMU_PLTRST_N)—Offset 678h" on page 3601 |
| 67C | 00000031 | "Pad Configuration DW1 (PAD_CFG_DW1_PMU_PLTRST_N)—Offset 67Ch" on page 3604 |
| 680 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SUS_STAT_N)—Offset 680h" on page 3605 |
| 684 | 00000032 | "Pad Configuration DW1 (PAD_CFG_DW1_SUS_STAT_N)—Offset 684h" on page 3608 |
| 688 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SLP_S0IX_N)—Offset 688h" on page 3609 |
| 68C | 00000033 | "Pad Configuration DW1 (PAD_CFG_DW1_SLP_S0IX_N)—Offset 68Ch" on page 3612 |
| 690 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS0_N)—Offset 690h" on page 3613 |
| 694 | 00003C34 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS0_N)—Offset 694h" on page 3616 |
| 698 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS1_N)—Offset 698h" on page 3617 |
| 69C | 00003C35 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS1_N)—Offset 69Ch" on page 3620 |
| 6A0 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_MOSI_IO0)—Offset 6A0h" on page 3621 |
| 6A4 | 00003C36 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_MOSI_IO0)—Offset 6A4h" on page 3624 |
| 6A8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_MISO_IO1)—Offset 6A8h" on page 3625 |
| 6AC | 00003C37 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_MISO_IO1)—Offset 6ACh" on page 3628 |
| 6B0 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO2)—Offset 6B0h" on page 3629 |
| 6B4 | 00003C38 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO2)—Offset 6B4h" on page 3632 |
| 6B8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO3)—Offset 6B8h" on page 3633 |
| 6BC | 00003C39 | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO3)—Offset 6BCh" on page 3636 |
| 6C0 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK)—Offset 6C0h" on page 3637 |
| 6C4 | 00003C3A | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK)—Offset 6C4h" on page 3640 |
| 6C8 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK_LOOPBK)—Offset 6C8h" on page 3641 |
| 6CC | 00003C3B | "Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK_LOOPBK)—Offset 6CCh" on page 3644 |
| 6D0 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO0)—Offset 6D0h" on page 3645 |
| 6D4 | 00003C3C | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO0)—Offset 6D4h" on page 3648 |
| 6D8 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO1)—Offset 6D8h" on page 3649 |
| 6DC | 00003C3D | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO1)—Offset 6DCh" on page 3652 |
| 6E0 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO2)—Offset 6E0h" on page 3653 |
| 6E4 | 00003C3E | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO2)—Offset 6E4h" on page 3656 |
| 6E8 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO3)—Offset 6E8h" on page 3657 |

## Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 8 of 9)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 6EC | 00003C3F | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO3)—Offset 6ECh" on page 3660 |
| 6F0 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CS0_N)—Offset 6F0h" on page 3661 |
| 6F4 | 00000040 | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CS0_N)—Offset 6F4h" on page 3664 |
| 6F8 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK)—Offset 6F8h" on page 3665 |
| 6FC | 00001041 | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK)—Offset 6FCh" on page 3668 |
| 700 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_RST_N)—Offset 700h" on page 3669 |
| 704 | 00001042 | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_RST_N)—Offset 704h" on page 3672 |
| 708 | 45000A00 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_ALRT0_N)—Offset 708h" on page 3673 |
| 70C | 00000043 | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_ALRT0_N)—Offset 70Ch" on page 3676 |
| 710 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_10)—Offset 710h" on page 3677 |
| 714 | 00000044 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_10)—Offset 714h" on page 3680 |
| 718 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_11)—Offset 718h" on page 3681 |
| 71C | 00000045 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_11)—Offset 71Ch" on page 3684 |
| 720 | 44000600 | "Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK_LOOPBK)—Offset 720h" on page 3685 |
| 724 | 00000046 | "Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK_LOOPBK)—Offset 724h" on page 3688 |
| 728 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CMD)—Offset 728h" on page 3689 |
| 72C | 00000047 | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CMD)—Offset 72Ch" on page 3692 |
| 730 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_STROBE)—Offset 730h" on page 3693 |
| 734 | 00000048 | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_STROBE)—Offset 734h" on page 3696 |
| 738 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CLK)—Offset 738h" on page 3697 |
| 73C | 00000049 | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CLK)—Offset 73Ch" on page 3700 |
| 740 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D0)—Offset 740h" on page 3701 |
| 744 | 0000004A | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D0)—Offset 744h" on page 3704 |
| 748 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D1)—Offset 748h" on page 3705 |
| 74C | 0000004B | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D1)—Offset 74Ch" on page 3708 |
| 750 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D2)—Offset 750h" on page 3709 |
| 754 | 0000004C | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D2)—Offset 754h" on page 3712 |
| 758 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D3)—Offset 758h" on page 3713 |
| 75C | 0000004D | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D3)—Offset 75Ch" on page 3716 |
| 760 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D4)—Offset 760h" on page 3717 |
| 764 | 0000004E | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D4)—Offset 764h" on page 3720 |
| 768 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D5)—Offset 768h" on page 3721 |
| 76C | 0000004F | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D5)—Offset 76Ch" on page 3724 |
| 770 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D6)—Offset 770h" on page 3725 |
| 774 | 00000050 | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D6)—Offset 774h" on page 3728 |
| 778 | 45000300 | "Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D7)—Offset 778h" on page 3729 |

**Table 65-2. Summary of Sideband Registers, Part 2 of 2—Sideband Port 0xC5 (Sheet 9 of 9)**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 77C | 00000051 | "Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D7)—Offset 77Ch" on page 3732 |
| 780 | 45000200 | "Pad Configuration DW0 (PAD_CFG_DW0_GPIO_3)—Offset 780h" on page 3733 |
| 784 | 00000052 | "Pad Configuration DW1 (PAD_CFG_DW1_GPIO_3)—Offset 784h" on page 3736 |

## 65.2 Sideband Registers, Part 1 of 2—Sideband Port 0xC2

### 65.2.1 Pad Base Address (PADBAR)—Offset Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + Ch

**Default:** 00000400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 15:0 | 0x400<br>RO | **Pad Base Address (PADBAR):**<br>This field provides the starting byte-align address of Pad0 register sets within a Community.<br>It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets. |

## 65.2.2 Miscellaneous Configuration (MISCCFG)—Offset 10h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 10h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 10h

**Default:** 00021000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Reserved (Reserved1):** Reserved |
| 19:16 | 0x2<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW2):**<br>This register assigns a specific GPIO Group to the ACPI GPE0[95:64].<br>The setting is SoC specific. |
| 15:12 | 0x1<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW1):**<br>This register assigns a specific GPIO Group to the ACPI GPE0[63:32].<br>The setting is SoC specific. |
| 11:8 | 0x0<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0[3:0]):<br>This register assigns a specific GPIO Group to the ACPI GPE0[31:0].<br>The setting is SoC specific. |
| 7:5 | 0x0<br>RW | **Reserved (Reserved2):** Reserved |
| 4 | 0x0<br>RW | **NCSI Outbound Signal Staging Control (NCSI_STAGE_EN):** NCSI Signal Staging Control:<br>This register controls the bypass/staging logic for the GBE NCSI output signals - when 1, the signals will be staged with NCSI_ARB_CLK; when 0, the flops will be bypassed. |
| 3 | 0x0<br>RW | **GPIO Driver IRQ Route (GPIO_DRIVER_IRQ_ROUTE):** GPIO Driver IRQ_ROUTE[1:0]:<br>Specifies the APIC IRQ globally for all pads within the current community (GPI_IS with corresponding GPI_IE enable).<br>• 0 = IRQ14<br>• 1 = IRQ15 |
| 2 | 0x0<br>RW | **GSX Static Local Clock Gating (GSXSLCGEN):** GSX Static Local Clock Gating (GSXSLCGEn)<br>Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available).<br>• 0 = Disable dynamic local clock gating<br>• 1 = Enable dynamic local clock gating |
| 1 | 0x0<br>RW | **GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** GPIO Dynamic Partition Clock Gating Enable (GPDPCGEn)<br>Specify whether the GPIO Community should take part in partition clock gating<br>• 0 = Disable participation in dynamic partition clock gating<br>• 1 = Enable participation in dynamic partition clock gating |
| 0 | 0x0<br>RW | **GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** GPIO Dynamic Local Clock Gating Enable (GPDLCGEn)<br>Specify whether the GPIO Community should perform local clock gating<br>• 0 = Disable dynamic local clock gating<br>• 1 = Enable dynamic local clock gating |

## 65.2.3 Pad Ownership (PAD_OWN_NORTH_ALL_0)—Offset 20h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 20h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 20h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_SDP3):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_SDP3):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_SDP2):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_SDP2):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_SDP1):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_SDP1):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_SDP0):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_SDP0):** See Description for bits [29:28] of this register. |

## 65.2.4 Pad Ownership (PAD_OWN_NORTH_ALL_1)—Offset 24h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 24h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 24h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_CLK_IN):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_RXD0):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_I2C_DATA):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_I2C_CLK):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_I2C_DATA):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_I2C_CLK):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE2_LED1):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE2_LED0):** See Description for bits [29:28] of this register. |

## 65.2.5 Pad Ownership (PAD_OWN_NORTH_ALL_2)—Offset 28h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 28h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_LED0):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_ARB_OUT):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_TXD1):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_TXD0):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_TX_EN):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_ARB_IN):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_CRS_DV):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_NCSI_RXD1):** See Description for bits [29:28] of this register. |

## 65.2.6 Pad Ownership (PAD_OWN_NORTH_ALL_3)—Offset 2Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 2Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ3_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ2_N):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ1_N):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ0_N):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_0):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_LED1):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE1_LED0):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GBE0_LED1):** See Description for bits [29:28] of this register. |

## 65.2.7 Pad Ownership (PAD_OWN_NORTH_ALL_4)—Offset 30h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 30h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PROCHOT_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_THERMTRIP_N):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SVID_CLK):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SVID_DATA):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SVID_ALERT_N):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_2):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_1):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ4_N):** See Description for bits [29:28] of this register. |

## 65.2.8 Pad Ownership (PAD_OWN_NORTH_ALL_5)—Offset 34h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 34h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 34h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:2 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_MEMHOT_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |

## 65.2.9 Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_0)—Offset 90h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 90h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ3_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_ARB_IN):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.10 Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_0)—Offset 94h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 94h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ3_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_ARB_IN):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.11 Pad Configuration Lock (PADCFGLOCK_NORTH_ALL_1)—Offset 98h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 98h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_MEMHOT_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.12 Pad Configuration Lock (PADCFGLOCKTX_NORTH_ALL_1)—Offset 9Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 9Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_MEMHOT_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.13 Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_0)—Offset C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + C0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ3_N):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_CLK_IN):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.14 Host Software Pad Ownership (HOSTSW_OWN_NORTH_ALL_1)—Offset C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + C4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_MEMHOT_N):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.15 GPI Interrupt Status (GPI_IS_NORTH_ALL_0)—Offset 100h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 100h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 100h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ3_N):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_CRS_DV):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.16 GPI Interrupt Status (GPI_IS_NORTH_ALL_1)—Offset 104h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 104h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_MEMHOT_N):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.17 GPI Interrupt Enable (GPI_IE_NORTH_ALL_0)—Offset 120h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 120h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ3_N):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_CRS_DV):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.18 GPI Interrupt Enable (GPI_IE_NORTH_ALL_1)—Offset 124h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 124h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_MEMHOT_N):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>Bit N-1= pad N-1 |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.19 Reserved[0-5] (RSVD6[0-5])—Offset 128h, Count 6, Stride 4h

## 65.2.20 GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_0)—Offset 140h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 140h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 140h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ3_N):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_ARB_IN):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.21 GPI General Purpose Events Status (GPI_GPE_STS_NORTH_ALL_1)—Offset 144h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 144h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 144h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_MEMHOT_N):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>Bit N-1= Pad N-1 |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.22 GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_0)—Offset 160h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 160h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 160h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ3_N):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_RXD1):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.23 GPI General Purpose Events Enable (GPI_GPE_EN_NORTH_ALL_1)—Offset 164h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 164h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 164h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_MEMHOT_N):** This bit is used to enable/ disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.24 SMI Status (GPI_SMI_STS_NORTH_ALL_0)—Offset 180h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 180h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 180h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ3_N):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 30 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_ARB_IN):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.25 SMI Status (GPI_SMI_STS_NORTH_ALL_1)—Offset 184h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 184h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 184h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_MEMHOT_N):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 7 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_1):**See Description for bit [8] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.26 SMI Enable (GPI_SMI_EN_NORTH_ALL_0)—Offset 1A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1A0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1A0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ3_N):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 30 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_RXD1):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.27 SMI Enable (GPI_SMI_EN_NORTH_ALL_1)—Offset 1A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1A4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1A4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_MEMHOT_N):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 7 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.28 NMI Status (GPI_NMI_STS_NORTH_ALL_0)—Offset 1C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1C0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ3_N):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_CRS_DV):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_RXD1):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.29 NMI Status (GPI_NMI_STS_NORTH_ALL_1)—Offset 1C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1C4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_MEMHOT_N):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 7 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.30 NMI Enable (GPI_NMI_EN_NORTH_ALL_0)—Offset 1E0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1E0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1E0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ3_N):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ2_N):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ1_N):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ0_N):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_0):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_LED1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_LED0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_LED1):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_LED0):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_ARB_OUT):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_TXD1):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_TXD0):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_TX_EN):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_ARB_IN):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_CRS_DV):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_RXD1):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_CLK_IN):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_NCSI_RXD0):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_I2C_DATA):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_I2C_CLK):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_I2C_DATA):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_I2C_CLK):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE2_LED1):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE2_LED0):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_SDP3):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_SDP3):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_SDP2):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_SDP2):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_SDP1):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_SDP1):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE1_SDP0):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_GBE0_SDP0):** See Description for bit [31] of this register. |

## 65.2.31 NMI Enable (GPI_NMI_EN_NORTH_ALL_1)—Offset 1E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 1E4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 1E4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:9 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 8 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_MEMHOT_N):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 7 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PROCHOT_N):** See Description for bit [8] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_THERMTRIP_N):** See Description for bit [8] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SVID_CLK):** See Description for bit [8] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SVID_DATA):** See Description for bit [8] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SVID_ALERT_N):** See Description for bit [8] of this register. |
| 2 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_2):** See Description for bit [8] of this register. |
| 1 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_1):** See Description for bit [8] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ4_N):** See Description for bit [8] of this register. |

## 65.2.32 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP0)—Offset 400h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 400h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 400h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.33 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP0)—Offset 404h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 404h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 404h

**Default:** 00000018h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x18<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.34 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP0)—Offset 408h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 408h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 408h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.35 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP0)—Offset 40Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 40Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 40Ch

**Default:** 00000019h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x19<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.36 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP1)—Offset 410h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 410h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 410h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.37 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP1)—Offset 414h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 414h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 414h

**Default:** 0000001Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.38 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP1)—Offset 418h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 418h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 418h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.39 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP1)—Offset 41Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 41Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 41Ch

**Default:** 0000001Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.40 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP2)—Offset 420h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 420h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 420h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.41 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP2)—Offset 424h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 424h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 424h

**Default:** 0000001Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.42 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP2)—Offset 428h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 428h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 428h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.43 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP2)—Offset 42Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 42Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 42Ch

**Default:** 0000001Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.44 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_SDP3)—Offset 430h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 430h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 430h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.45 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_SDP3)—Offset 434h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 434h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 434h

**Default:** 0000001Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.46 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_SDP3)—Offset 438h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 438h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 438h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.47 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_SDP3)—Offset 43Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 43Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 43Ch

**Default:** 0000001Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.48 Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED0)—Offset 440h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 440h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 440h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.49 Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED0)—Offset 444h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 444h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 444h

**Default:** 00000020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x20<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.50 Pad Configuration DW0 (PAD_CFG_DW0_GBE2_LED1)—Offset 448h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 448h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 448h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.51 Pad Configuration DW1 (PAD_CFG_DW1_GBE2_LED1)—Offset 44Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 44Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 44Ch

**Default:** 00000021h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x21<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.52 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_CLK)—Offset 450h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 450h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 450h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.53 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_CLK)—Offset 454h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 454h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 454h

**Default:** 00002822h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xa<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x22<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.54 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_I2C_DATA)—Offset 458h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 458h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 458h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.55 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_I2C_DATA)—Offset 45Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 45Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 45Ch

**Default:** 00002823h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xa<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x23<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.56 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_CLK)—Offset 460h

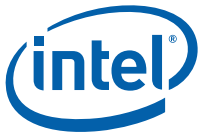
**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 460h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 460h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |



| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.57 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_CLK)—Offset 464h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 464h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 464h

**Default:** 00002824h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xa<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x24<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.58 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_I2C_DATA)—Offset 468h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 468h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 468h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.59 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_I2C_DATA)—Offset 46Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 46Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 46Ch

**Default:** 00002825h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xa<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x25<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.60 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD0)—Offset 470h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 470h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 470h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.61 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD0)—Offset 474h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 474h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 474h

**Default:** 00003026h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x26<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.62 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CLK_IN)—Offset 478h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 478h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 478h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.63 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CLK_IN)—Offset 47Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 47Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 47Ch

**Default:** 00003027h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x27<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.64 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_RXD1)—Offset 480h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 480h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 480h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.65 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_RXD1)—Offset 484h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 484h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 484h

**Default:** 00003028h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x28<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.66 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_CRS_DV)—Offset 488h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 488h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 488h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.67 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_CRS_DV)—Offset 48Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 48Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 48Ch

**Default:** 00000029h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x29<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.68 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_IN)—Offset 490h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 490h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 490h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.69 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_IN)—Offset 494h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 494h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 494h

**Default:** 0000102Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x4<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.70 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TX_EN)—Offset 498h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 498h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 498h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.71 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TX_EN)—Offset 49Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 49Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 49Ch

**Default:** 0000002Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.72 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD0)—Offset 4A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4A0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4A0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.73 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD0)—Offset 4A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4A4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4A4h

**Default:** 0000002Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.74 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_TXD1)—Offset 4A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4A8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4A8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.75 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_TXD1)—Offset 4ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4ACh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4ACh

**Default:** 0000002Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.76 Pad Configuration DW0 (PAD_CFG_DW0_NCSI_ARB_OUT)—Offset 4B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4B0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4B0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.77 Pad Configuration DW1 (PAD_CFG_DW1_NCSI_ARB_OUT)—Offset 4B4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4B4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4B4h

**Default:** 0000002Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.78 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED0)—Offset 4B8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4B8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4B8h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.79 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED0)—Offset 4BCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4BCh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4BCh

**Default:** 0000002Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.80 Pad Configuration DW0 (PAD_CFG_DW0_GBE0_LED1)—Offset 4C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4C0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4C0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.81 Pad Configuration DW1 (PAD_CFG_DW1_GBE0_LED1)—Offset 4C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4C4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4C4h

**Default:** 00000030h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x30<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.82 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED0)—Offset 4C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4C8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4C8h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.83 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED0)—Offset 4CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4CCh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4CCh

**Default:** 00000031h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x31<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.84 Pad Configuration DW0 (PAD_CFG_DW0_GBE1_LED1)—Offset 4D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4D0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4D0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.85 Pad Configuration DW1 (PAD_CFG_DW1_GBE1_LED1)—Offset 4D4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4D4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4D4h

**Default:** 00000032h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x32<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.86 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_0)—Offset 4D8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4D8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4D8h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.87 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_0)—Offset 4DCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4DCh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4DCh

**Default:** 00000033h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x33<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.88 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ0_N)—Offset 4E0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4E0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4E0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.89 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ0_N)—Offset 4E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4E4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4E4h

**Default:** 00000034h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x34<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.90 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ1_N)—Offset 4E8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4E8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4E8h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.91 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ1_N)—Offset 4ECh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4ECh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4ECh

**Default:** 00000035h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x35<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.92 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ2_N)—Offset 4F0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4F0h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4F0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.93 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ2_N)—Offset 4F4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4F4h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4F4h

**Default:** 00000036h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x36<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.94 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ3_N)—Offset 4F8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4F8h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4F8h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.95 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ3_N)—Offset 4FCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 4FCh
**Host Memory Space:** SBREG_BAR + 0xC20000 + 4FCh

**Default:** 00000037h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x37<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.96 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ4_N)—Offset 500h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 500h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 500h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.97 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ4_N)—Offset 504h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 504h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 504h

**Default:** 00000038h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x38<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.98 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_1)—Offset 508h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 508h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 508h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Pre Glitch Filter Stage RX Pad State Select (PreGfRXSel) Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.99 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_1)—Offset 50Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 50Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 50Ch

**Default:** 00000039h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x39<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.100 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_2)—Offset 510h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 510h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 510h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.101 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_2)—Offset 514h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 514h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 514h

**Default:** 0000003Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.102 Pad Configuration DW0 (PAD_CFG_DW0_SVID_ALERT_N)—Offset 518h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 518h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 518h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.103 Pad Configuration DW1 (PAD_CFG_DW1_SVID_ALERT_N)—Offset 51Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 51Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 51Ch

**Default:** 0000003Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.104 Pad Configuration DW0 (PAD_CFG_DW0_SVID_DATA)—Offset 520h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 520h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 520h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.105 Pad Configuration DW1 (PAD_CFG_DW1_SVID_DATA)—Offset 524h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 524h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 524h

**Default:** 0000003Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.106 Pad Configuration DW0 (PAD_CFG_DW0_SVID_CLK)—Offset 528h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 528h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 528h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.107 Pad Configuration DW1 (PAD_CFG_DW1_SVID_CLK)—Offset 52Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 52Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 52Ch

**Default:** 0000003Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.108 Pad Configuration DW0 (PAD_CFG_DW0_THERMTRIP_N)—Offset 530h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 530h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 530h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.109 Pad Configuration DW1 (PAD_CFG_DW1_THERMTRIP_N)—Offset 534h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 534h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 534h

**Default:** 0000303Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.110 Pad Configuration DW0 (PAD_CFG_DW0_PROCHOT_N)—Offset 538h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 538h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 538h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.111 Pad Configuration DW1 (PAD_CFG_DW1_PROCHOT_N)—Offset 53Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 53Ch
**Host Memory Space:** SBREG_BAR + 0xC20000 + 53Ch

**Default:** 0000303Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.2.112 Pad Configuration DW0 (PAD_CFG_DW0_MEMHOT_N)—Offset 540h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 540h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 540h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.2.113 Pad Configuration DW1 (PAD_CFG_DW1_MEMHOT_N)—Offset 544h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC2] + 544h
**Host Memory Space:** SBREG_BAR + 0xC20000 + 544h

**Default:** 00003040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x40<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

# 65.3 Sideband Registers, Part 2 of 2—Sideband Port 0xC5

## 65.3.1 Pad Base Address (PADBAR)—Offset 0Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 0Ch

**Default:** 00000400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:16 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved |
| 15:0 | 0x400<br>RO | **Pad Base Address (PADBAR):** This field provides the starting byte-align address of Pad0 register sets within a Community.<br>It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets. |

## 65.3.2 Miscellaneous Configuration (MISCCFG)—Offset 10h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 10h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 10h

**Default:** 00021000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:20 | 0x0<br>RW | **Reserved (RSVD_1):** Reserved |
| 19:16 | 0x2<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64]. |
| 15:12 | 0x1<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW1):** GPIO Group to GPE_DW0 assignment encoding (GPE0_DW1[3:0]) : This register assigns a specific GPIO Group to the ACPI GPE0[63:32]. |
| 11:8 | 0x0<br>RW | **GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0[3:0]) : This register assigns a specific GPIO Group to the ACPI GPE0[31:0]. |
| 7:5 | 0x0<br>RW | **Reserved (Re2):** Reserved |
| 4 | 0x0<br>RW | **NCSI Inbound Signal Staging Control (NCSI_STAGE_EN):** This register controls the bypass/ staging logic for the GBE NCSI input signals - when 1, the signals will be staged with NCSI_ARB_CLK; when 0, the flops will be bypassed. |
| 3 | 0x0<br>RW | **GPIO Driver IRQ Route (GPIO_DRIVER_IRQ_ROUTE):** GPIO Driver IRQ_ROUTE[1:0]: Specifies the APIC IRQ globally for all pads within the current community (GPI_IS with corresponding GPI_IE enable).<br>• 0 = IRQ14<br>• 1 = IRQ15 |
| 2 | 0x0<br>RW | **GSX Static Local Clock Gating (GSXSLCGEN):** GSX Static Local Clock Gating (GSXSLCGEn)<br>Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available).<br>• 0 = Disable dynamic local clock gating<br>• 1 = Enable dynamic local clock gating |
| 1 | 0x0<br>RW | **GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** GPIO Dynamic Partition Clock Gating Enable (GPDPCGEn)<br>Specify whether the GPIO Community should take part in partition clock gating<br>• 0 = Disable participation in dynamic partition clock gating<br>• 1 = Enable participation in dynamic partition clock gating |
| 0 | 0x0<br>RW | **GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** GPIO Dynamic Local Clock Gating Enable (GPDLCGEn)<br>Specify whether the GPIO Community should perform local clock gating.<br>• 0 = Disable dynamic local clock gating<br>• 1 = Enable dynamic local clock gating |

## 65.3.3 Pad Ownership (PAD_OWN_SOUTH_DFX_0)—Offset 20h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 20h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 20h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT5):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT4):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT3):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT2):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT1):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT0):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT_CLK1):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT_CLK0):** See Description for bits [29:28] of this register. |

## 65.3.4 Pad Ownership (PAD_OWN_SOUTH_DFX_1)—Offset 24h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 24h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 24h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT13):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT12):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT11):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT10):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT9):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT8):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT7):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT6):** See Description for bits [29:28] of this register. |

## 65.3.5 Pad Ownership (PAD_OWN_SOUTH_DFX_2)—Offset 28h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 28h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 28h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT15):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_PORT14):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |

## 65.3.6 Pad Ownership (PAD_OWN_SOUTH_GROUP0_0)—Offset 2Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 2Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 2Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB5_GBE_CLK):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_UART0_TXD):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_UART0_RXD):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ7_N):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ6_N):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PCIE_CLKREQ5_N):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB5_GBE_ALRT_N):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_12):** See Description for bits [29:28] of this register. |

## 65.3.7 Pad Ownership (PAD_OWN_SOUTH_GROUP0_1)—Offset 30h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 30h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 30h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB0_LEG_DATA):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB0_LEG_CLK):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_MCERR_N):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_IERR_N):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ERROR0_N):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ERROR1_N):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ERROR2_N):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB5_GBE_DATA):** See Description for bits [29:28] of this register. |

## 65.3.8 Pad Ownership (PAD_OWN_SOUTH_GROUP0_2)—Offset 34h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 34h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 34h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB4_CSME0_ALRT_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB4_CSME0_CLK):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB4_CSME0_DATA):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB2_PECI_CLK):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB2_PECI_DATA):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB1_HOST_CLK):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB1_HOST_DATA):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SMB0_LEG_ALRT_N):** See Description for bits [29:28] of this register. |

## 65.3.9 Pad Ownership (PAD_OWN_SOUTH_GROUP0_3)—Offset 38h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 38h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 38h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA0_LED_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_7):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_6):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_5):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_4):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_FLEX_CLK_SE1):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_FLEX_CLK_SE0):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_USB_OC0_N):** See Description for bits [29:28] of this register. |

## 65.3.10 Pad Ownership (PAD_OWN_SOUTH_GROUP0_4)—Offset 3Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 3Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 3Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_8):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_UART1_TXD):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_UART1_RXD):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA1_SDOUT):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA0_SDOUT):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA_PDETECT1):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA_PDETECT0):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SATA1_LED_N):** See Description for bits [29:28] of this register. |

## 65.3.11 Pad Ownership (PAD_OWN_SOUTH_GROUP0_5)—Offset 40h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 40h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 40h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_CX_PREQ_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_CX_PRDY_N):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_TDO):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_TDI):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_TMS):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_TRST_N):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_TCK):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_9):** See Description for bits [29:28] of this register. |

## 65.3.12 Pad Ownership (PAD_OWN_SOUTH_GROUP0_6)—Offset 44h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 44h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 44h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_SPARE4):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_SPARE3):** See Description for bits [17:16] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_DFX_SPARE2):** See Description for bits [17:16] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_CTBTRIGOUT):** See Description for bits [17:16] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_CTBTRIGINOUT):** See Description for bits [17:16] of this register. |

## 65.3.13 Pad Ownership (PAD_OWN_SOUTH_GROUP1_0)—Offset 48h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 48h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 48h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_RESETBUTTON_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_PWRBTN_N):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_WAKE_N):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_SLP_S3_N):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_SLP_S45_N):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ADR_TRIGGER):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_SUSCLK):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SUSPWRDNACK):** See Description for bits [29:28] of this register. |

## 65.3.14 Pad Ownership (PAD_OWN_SOUTH_GROUP1_1)—Offset 4Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_IO2):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_MISO_IO1):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_MOSI_IO0):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_CS1_N):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_CS0_N):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SLP_S0IX_N):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SUS_STAT_N):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_PMU_PLTRST_N):** See Description for bits [29:28] of this register. |

## 65.3.15 Pad Ownership (PAD_OWN_SOUTH_GROUP1_2)—Offset 50h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 50h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 50h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_CS0_N):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_IO3):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_IO2):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_IO1):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_IO0):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_CLK_LOOPBK):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_CLK):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_SPI_IO3):** See Description for bits [29:28] of this register. |

## 65.3.16 Pad Ownership (PAD_OWN_SOUTH_GROUP1_3)—Offset 54h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 54h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 54h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_STROBE):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_CMD):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_CLK_LOOPBK):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_11):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_10):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_ALRT0_N):** See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_RST_N):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_ESPI_CLK):** See Description for bits [29:28] of this register. |

## 65.3.17 Pad Ownership (PAD_OWN_SOUTH_GROUP1_4)—Offset 58h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 58h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 58h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 29:28 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D6):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 27:26 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 25:24 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D5):** See Description for bits [29:28] of this register. |
| 23:22 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 21:20 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D4):** See Description for bits [29:28] of this register. |
| 19:18 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved. |
| 17:16 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D3):** See Description for bits [29:28] of this register. |
| 15:14 | 0x0<br>RO | **Reserved (RSVD_4):** Reserved. |
| 13:12 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D2):** See Description for bits [29:28] of this register. |
| 11:10 | 0x0<br>RO | **Reserved (RSVD_5):** Reserved. |
| 9:8 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D1):**See Description for bits [29:28] of this register. |
| 7:6 | 0x0<br>RO | **Reserved (RSVD_6):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D0):** See Description for bits [29:28] of this register. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_7):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_CLK):** See Description for bits [29:28] of this register. |

## 65.3.18 Pad Ownership (PAD_OWN_SOUTH_GROUP1_5)—Offset 5Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 5:4 | 0x0<br>RW | **Pad Ownership (PAD_OWN_GPIO_3):** This register can be written by the Intel® Management Engine (Intel® ME) GPIO Function only.** Write access without Intel® ME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Encoding:<br>• 00 = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_INT_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, Intel® ME, IE and ISH do not own this pad and are not notified of the GPIO input event.<br>• 01 = Intel® ME GPIO Mode. The Intel® ME has ownership of the pad. All accesses to Pad Configuration register set must be verified with Intel® ME GPIO Function SAI **. Non-posted cycle access without Intel® ME SAI shall result in Unsuccessful Completion status. Posted cycle access without Intel® ME SAI shall be dropped. In Intel® ME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to Intel® ME GPIO Function. Only the Intel® ME GPIO Function is notified of the GPIO input event.<br>• 10 = ISH GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function.<br>• 11 = IE GPIO Mode. Same description as Intel® ME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the IE GPIO Function.<br>**During DFx mode, Secure DFx can write to the PAD_OWN. |
| 3:2 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 1:0 | 0x0<br>RW | **Pad Ownership (PAD_OWN_EMMC_D7):** See Description for bits [5:4] of this register. |

## 65.3.19 Pad Configuration Lock (PADCFGLOCK_SOUTH_DFX_0)—Offset 90h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 90h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 90h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT15):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 16 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT3):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT2):** See Description for bit [17] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT1):** See Description for bit [17] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

## 65.3.20 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_DFX_0)—Offset 94h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 94h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 94h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT15):** Pad Configuration Lock GPIO TxState (PadCfgLockTx) PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1 = pad N-1<br>When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 16 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT2):** See Description for bit [17] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT1):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

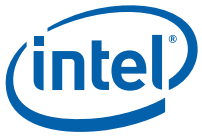
## 65.3.21 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_0)—Offset 98h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 98h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 98h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA0_LED_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB4_CSME0_CLK):**See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB1_HOST_CLK):** See Description for bit [31] of this register. |



| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.22 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_0)—Offset 9Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 9Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 9Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA0_LED_N):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB1_HOST_CLK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.23 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP0_1)—Offset A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + A0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + A0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_SPARE4):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 19 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_9):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.24 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP0_1)—Offset A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + A4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + A4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_SPARE4):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 19 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_9):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.25 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_0)—Offset A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + A8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + A8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_STROBE):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_SUSPWRDNACK):** See Description for bit [31] of this register. |

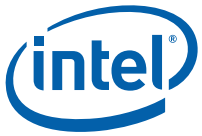
## 65.3.26 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_0)—Offset ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + ACh
**Host Memory Space:** SBREG_BAR + 0xC50000 + ACh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_STROBE):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 30 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |



| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.27 Pad Configuration Lock (PADCFGLOCK_SOUTH_GROUP1_1)—Offset B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + B0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_GPIO_3):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 8 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock (PADCFGLOCK_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.28 Pad Configuration Lock (PADCFGLOCKTX_SOUTH_GROUP1_1)—Offset B4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + B4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + B4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_GPIO_3):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect.<br>This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock.<br>0 = Unlock<br>1= Lock the following register fields as read-only (RO):<br>• Pad Configuration registers (exclude GPIOTXState)<br>• GPI_NMI_EN Register (if implemented)<br>• GPI_SMI_EN Register (if implemented)<br>• GPI_GPE_EN Register (if implemented)<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. |
| 8 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **Pad Config Lock TXState (PADCFGLOCKTX_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.29 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_DFX_0)—Offset C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + C0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + C0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT15):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 16 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT2):** See Description for bit [17] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT1):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

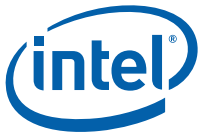
## 65.3.30 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_0)—Offset C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + C4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA0_LED_N):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |



| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.31 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP0_1)—Offset C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + C8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_SPARE4):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 19 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_UART1_TXD):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.32 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_0)—Offset CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + CCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_STROBE):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_IO3):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.33 Host Software Pad Ownership (HOSTSW_OWN_SOUTH_GROUP1_1)—Offset D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + D0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_GPIO_3):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN).<br>0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_INT_STS update is masked.<br>1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_INT_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked.<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 8 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **HostSW_Own (HOSTSW_OWN_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.34 GPI Interrupt Status (GPI_IS_SOUTH_DFX_0)—Offset 100h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 100h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 100h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT15):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 16 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT2):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT1):** See Description for bit [17] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

## 65.3.35 GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_0)—Offset 104h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 104h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 104h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA0_LED_N):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB1_HOST_DATA):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.36 GPI Interrupt Status (GPI_IS_SOUTH_GROUP0_1)—Offset 108h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 108h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 108h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI Interrupt Status (GPI_INT_STS_DFX_SPARE4):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 19 | 0x0<br>RO | **GPI Interrupt Status (GPI_INT_STS_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI Interrupt Status (GPI_INT_STS_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_8):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.37 GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_0)—Offset 10Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 10Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 10Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_STROBE):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_CLK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.38 GPI Interrupt Status (GPI_IS_SOUTH_GROUP1_1)—Offset 110h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 110h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 110h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_GPIO_3):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode<br>• The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode).<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = No interrupt<br>• 1 = Interrupt asserts<br>The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x].<br>• Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 8 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW/1C | **GPI Interrupt Status (GPI_INT_STS_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.39 GPI Interrupt Enable (GPI_IE_SOUTH_DFX_0)—Offset 120h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 120h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT15):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set.<br>• 0 = disable interrupt generation<br>• 1 = enable interrupt generation<br>Refer to GPIO_DRIVER_IRQ_ROUTE for host GPIO Driver Mode interrupt routing.<br>Each bit position correspond to one pad:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 16 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT2):** See Description for bit [17] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT1):** See Description for bit [17] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT0):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

## 65.3.40 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_0)—Offset 124h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 124h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 124h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA0_LED_N):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set.<br>• 0 = disable interrupt generation<br>• 1 = enable interrupt generation<br>Refer to GPIO_DRIVER_IRQ_ROUTE for host GPIO Driver Mode interrupt routing.<br>Each bit position correspond to one pad:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB0_LEG_DATA):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.41 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP0_1)—Offset 128h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 128h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 128h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI Interrupt Enable (GPI_INT_EN_DFX_SPARE4):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set.<br>• 0 = disable interrupt generation<br>• 1 = enable interrupt generation<br>Refer to GPIO_DRIVER_IRQ_ROUTE for host GPIO Driver Mode interrupt routing.<br>Each bit position correspond to one pad:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 19 | 0x0<br>RO | **GPI Interrupt Enable (GPI_INT_EN_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI Interrupt Enable (GPI_INT_EN_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_UART1_RXD):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.42 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_0)—Offset 12Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 12Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 12Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_STROBE):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set.<br>• 0 = disable interrupt generation<br>• 1 = enable interrupt generation<br>Refer to GPIO_DRIVER_IRQ_ROUTE for host GPIO Driver Mode interrupt routing.<br>Each bit position correspond to one pad:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 30 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_IO2):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 14 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.43 GPI Interrupt Enable (GPI_IE_SOUTH_GROUP1_1)—Offset 130h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 130h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 130h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_GPIO_3):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set.<br>• 0 = disable interrupt generation<br>• 1 = enable interrupt generation<br>Refer to GPIO_DRIVER_IRQ_ROUTE for host GPIO Driver Mode interrupt routing.<br>Each bit position correspond to one pad:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1 |
| 8 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **GPI Interrupt Enable (GPI_INT_EN_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.44 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_DFX_0)—Offset 140h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 140h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 140h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT15):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 16 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT2):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT1):** See Description for bit [17] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

## 65.3.45 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_0)—Offset 144h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 144h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 144h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA0_LED_N):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB1_HOST_DATA):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.46 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP0_1)—Offset 148h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 148h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 148h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_SPARE4):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 19 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_8):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.47 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_0)—Offset 14Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 14Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 14Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_STROBE):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_CLK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.48 GPI General Purpose Events Status (GPI_GPE_STS_SOUTH_GROUP1_1)—Offset 150h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 150h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 150h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_GPIO_3):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set:<br>• If the system is in an S3-S5 state, the event will also wake the system.<br>• If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position.<br>The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i].<br>Each bit position correspond to 1 pad in the Community:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 8 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW/1C | **GPI General Purpose Events Status (GPI_GPE_STS_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.49 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_DFX_0)—Offset 160h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 160h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 160h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:18 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 17 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT15):** This bit is used to enable/ disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 16 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT14):** See Description for bit [17] of this register. |
| 15 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT13):** See Description for bit [17] of this register. |
| 14 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT12):** See Description for bit [17] of this register. |
| 13 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT11):** See Description for bit [17] of this register. |
| 12 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT10):** See Description for bit [17] of this register. |
| 11 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT9):** See Description for bit [17] of this register. |
| 10 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT8):** See Description for bit [17] of this register. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT7):** See Description for bit [17] of this register. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT6):** See Description for bit [17] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT5):** See Description for bit [17] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT4):** See Description for bit [17] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT3):** See Description for bit [17] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT2):** See Description for bit [17] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT1):** See Description for bit [17] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT0):** See Description for bit [17] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT_CLK1):** See Description for bit [17] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_PORT_CLK0):** See Description for bit [17] of this register. |

## 65.3.50 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_0)—Offset 164h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 164h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 164h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA0_LED_N):** This bit is used to enable/ disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.51 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP0_1)—Offset 168h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 168h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 168h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_SPARE4):** This bit is used to enable/ disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 19 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_UART1_TXD):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.52 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_0)—Offset 16Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 16Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 16Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_STROBE):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 30 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_IO3):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.53 GPI General Purpose Events Enable (GPI_GPE_EN_SOUTH_GROUP1_1)—Offset 170h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 170h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 170h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_3):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set.<br>• 0 = disable GPE generation<br>• 1 = enable GPE generation<br>***Note:*** Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'.<br>Bit assignment:<br>• Bit0 = Pad0<br>• Bit1 = Pad1<br>• Bit2 = Pad2<br>• ...<br>• Bit N-1= Pad N-1 |
| 8 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **GPI General Purpose Events Enable (GPI_GPE_EN_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.54 SMI Status (GPI_SMI_STS_SOUTH_GROUP0_0)—Offset 184h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 184h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 184h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA0_LED_N):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 30 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB1_HOST_CLK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.55 SMI Status (GPI_SMI_STS_SOUTH_GROUP0_1)—Offset 188h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 188h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 188h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_DFX_SPARE4):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 19 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_9):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.56 SMI Status (GPI_SMI_STS_SOUTH_GROUP1_0)—Offset 18Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 18Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 18Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_STROBE):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 30 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 17 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Status (GPI_SMI_STS_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.57 SMI Status (GPI_SMI_STS_SOUTH_GROUP1_1)—Offset 190h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 190h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 190h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_GPIO_3):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode).<br>If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set:<br>1. The corresponding bit in the GPI_SMI_EN register is set<br>2. The corresponding pad's GPIROUTSMI is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no SMI event<br>• 1 = There is an SMI event<br>The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x].<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>Defaults for these bits are dependent on the state of the GPI pads. |
| 8 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW/1C | **GPI SMI Status (GPI_SMI_STS_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.58 SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_0)—Offset 1A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1A4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1A4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA0_LED_N):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 30 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.59 SMI Enable (GPI_SMI_EN_SOUTH_GROUP0_1)—Offset 1A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1A8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1A8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_DFX_SPARE4):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 19 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_UART1_TXD):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.60 SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_0)—Offset 1ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1ACh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1ACh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_STROBE):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 30 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_IO3):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_IO2):**See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI SMI Enable (GPI_SMI_EN_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.61 SMI Enable (GPI_SMI_EN_SOUTH_GROUP1_1)—Offset 1B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1B0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1B0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_GPIO_3):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'.<br>• 0 = disable SMI generation<br>• 1 = enable SMI generation<br>***Note:*** Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1 |
| 8 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **GPI SMI Enable (GPI_SMI_EN_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.62 NMI Status (GPI_NMI_STS_SOUTH_GROUP0_0)—Offset 1C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1C4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1C4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA0_LED_N):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB1_HOST_CLK):**See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB1_HOST_DATA):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.63 NMI Status (GPI_NMI_STS_SOUTH_GROUP0_1)—Offset 1C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1C8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1C8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_DFX_SPARE4):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 19 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_8):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 6 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_UART1_TXD):** See Description for bit [20] of this register. |
| 5 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.64 NMI Status (GPI_NMI_STS_SOUTH_GROUP1_0)—Offset 1CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1CCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1CCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_STROBE):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_CLK):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 16 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_IO3):** See Description for bit [31] of this register. |
| 15 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Status (GPI_NMI_STS_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.65 NMI Status (GPI_NMI_STS_SOUTH_GROUP1_1)—Offset 1D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1D0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1D0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_GPIO_3):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true:<br>• The corresponding pad is used in GPIO input mode (PMode)<br>• The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode).<br>• The corresponding GPI_NMI_EN is set<br>Writing a value of '1' will clear the bit while writing a value of '0' has no effect.<br>• 0 = There is no NMI event<br>• 1 = There is an NMI event<br>Bit assignment:<br>• Bit0 = pad 0<br>• Bit1 = pad 1<br>• Bit2 = pad 2<br>• ...<br>• Bit N-1= pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 8 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW/1C | **GPI NMI Status (GPI_NMI_STS_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.66 NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_0)—Offset 1E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1E4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1E4h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA0_LED_N):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_7):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_6):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_5):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_4):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_FLEX_CLK_SE1):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_FLEX_CLK_SE0):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_USB_OC0_N):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB4_CSME0_ALRT_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB4_CSME0_CLK):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB4_CSME0_DATA):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB2_PECI_CLK):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB2_PECI_DATA):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB1_HOST_CLK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB1_HOST_DATA):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB0_LEG_ALRT_N):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB0_LEG_DATA):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SMB0_LEG_CLK):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_MCERR_N):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_IERR_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ERROR0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ERROR1_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ERROR2_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SMB5_GBE_DATA):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SMB5_GBE_CLK):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_UART0_TXD):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_UART0_RXD):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ7_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ6_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_PCIE_CLKREQ5_N):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SMB5_GBE_ALRT_N):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_12):** See Description for bit [31] of this register. |

## 65.3.67 NMI Enable (GPI_NMI_EN_SOUTH_GROUP0_1)—Offset 1E8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1E8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1E8h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 20 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_DFX_SPARE4):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 19 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_DFX_SPARE3):** See Description for bit [20] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_DFX_SPARE2):** See Description for bit [20] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_CTBTRIGOUT):** See Description for bit [20] of this register. |
| 16 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_CTBTRIGINOUT):** See Description for bit [20] of this register. |
| 15 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_CX_PREQ_N):** See Description for bit [20] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_CX_PRDY_N):** See Description for bit [20] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_TDO):** See Description for bit [20] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_TDI):** See Description for bit [20] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_TMS):** See Description for bit [20] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_TRST_N):** See Description for bit [20] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_TCK):** See Description for bit [20] of this register. |
| 8 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_9):** See Description for bit [20] of this register. |
| 7 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_8):** See Description for bit [20] of this register. |
| 6 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_UART1_TXD):** See Description for bit [20] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 5 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_UART1_RXD):** See Description for bit [20] of this register. |
| 4 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA1_SDOUT):** See Description for bit [20] of this register. |
| 3 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA0_SDOUT):** See Description for bit [20] of this register. |
| 2 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA_PDETECT1):** See Description for bit [20] of this register. |
| 1 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA_PDETECT0):** See Description for bit [20] of this register. |
| 0 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_SATA1_LED_N):** See Description for bit [20] of this register. |

## 65.3.68 NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_0)—Offset 1ECh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1ECh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1ECh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_STROBE):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 30 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_CMD):** See Description for bit [31] of this register. |
| 29 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 28 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_11):** See Description for bit [31] of this register. |
| 27 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_10):** See Description for bit [31] of this register. |
| 26 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_ESPI_ALRT0_N):** See Description for bit [31] of this register. |
| 25 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_RST_N):** See Description for bit [31] of this register. |
| 24 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_CLK):** See Description for bit [31] of this register. |
| 23 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_CS0_N):** See Description for bit [31] of this register. |
| 22 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_IO3):** See Description for bit [31] of this register. |
| 21 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_IO2):** See Description for bit [31] of this register. |
| 20 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_IO1):** See Description for bit [31] of this register. |
| 19 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_ESPI_IO0):** See Description for bit [31] of this register. |
| 18 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_CLK_LOOPBK):** See Description for bit [31] of this register. |
| 17 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_CLK):** See Description for bit [31] of this register. |
| 16 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_IO3):** See Description for bit [31] of this register. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_IO2):** See Description for bit [31] of this register. |
| 14 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_MISO_IO1):** See Description for bit [31] of this register. |
| 13 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_MOSI_IO0):** See Description for bit [31] of this register. |
| 12 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_CS1_N):** See Description for bit [31] of this register. |
| 11 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SPI_CS0_N):** See Description for bit [31] of this register. |
| 10 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SLP_S0IX_N):** See Description for bit [31] of this register. |
| 9 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SUS_STAT_N):** See Description for bit [31] of this register. |
| 8 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_PLTRST_N):** See Description for bit [31] of this register. |
| 7 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_RESETBUTTON_N):** See Description for bit [31] of this register. |
| 6 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_PWRBTN_N):** See Description for bit [31] of this register. |
| 5 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_WAKE_N):** See Description for bit [31] of this register. |
| 4 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_SLP_S3_N):** See Description for bit [31] of this register. |
| 3 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_SLP_S45_N):** See Description for bit [31] of this register. |
| 2 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_ADR_TRIGGER):** See Description for bit [31] of this register. |
| 1 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_PMU_SUSCLK):** See Description for bit [31] of this register. |
| 0 | 0x0<br>RO | **GPI NMI Enable (GPI_NMI_EN_SUSPWRDNACK):** See Description for bit [31] of this register. |

## 65.3.69 NMI Enable (GPI_NMI_EN_SOUTH_GROUP1_1)—Offset 1F0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 1F0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 1F0h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:10 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 9 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_GPIO_3):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set.<br>• 0 = disable NMI generation<br>• 1 = enable NMI generation<br>Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only.<br>Bit assignment:<br>• Bit0 = Pad 0<br>• Bit1 = Pad 1<br>• Bit2 = Pad 2<br>• ...<br>• Bit N-1 = Pad N-1<br>For pads which do not support NMI, the corresponding bit is read-only zero. |
| 8 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D7):** See Description for bit [9] of this register. |
| 7 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D6):** See Description for bit [9] of this register. |
| 6 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D5):** See Description for bit [9] of this register. |
| 5 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D4):** See Description for bit [9] of this register. |
| 4 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D3):** See Description for bit [9] of this register. |
| 3 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D2):** See Description for bit [9] of this register. |
| 2 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D1):** See Description for bit [9] of this register. |
| 1 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_D0):** See Description for bit [9] of this register. |
| 0 | 0x0<br>RW | **GPI NMI Enable (GPI_NMI_EN_EMMC_CLK):** See Description for bit [9] of this register. |

## 65.3.70 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK0)—Offset 400h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 400h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 400h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>***Note:*** The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.71 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK0)—Offset 404h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 404h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 404h

**Default:** 00000041h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x41<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>• Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.72 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT_CLK1)—Offset 408h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 408h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 408h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.73 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT_CLK1)—Offset 40Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 40Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 40Ch

**Default:** 00000042h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x42<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.74 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT0)—Offset 410h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 410h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 410h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2 (PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1 (PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.75 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT0)—Offset 414h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 414h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 414h

**Default:** 00000043h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x43<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.76 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT1)—Offset 418h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 418h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 418h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.77 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT1)—Offset 41Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 41Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 41Ch

**Default:** 00000044h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x44<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.78 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT2)—Offset 420h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 420h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 420h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.79 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT2)—Offset 424h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 424h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 424h

**Default:** 00000045h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x45<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.80 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT3)—Offset 428h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 428h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 428h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.81 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT3)—Offset 42Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 42Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 42Ch

**Default:** 00000046h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x46<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.82 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT4)—Offset 430h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 430h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 430h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.83 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT4)—Offset 434h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 434h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 434h

**Default:** 00000047h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x47<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.84 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT5)—Offset 438h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 438h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 438h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.85 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT5)—Offset 43Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 43Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 43Ch

**Default:** 00000048h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x48<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.86 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT6)—Offset 440h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 440h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 440h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.87 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT6)—Offset 444h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 444h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 444h

**Default:** 00000049h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x49<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.88 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT7)—Offset 448h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 448h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 448h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.89 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT7)—Offset 44Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 44Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 44Ch

**Default:** 0000004Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.90 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT8)—Offset 450h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 450h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 450h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.91 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT8)—Offset 454h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 454h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 454h

**Default:** 0000004Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.92 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT9)—Offset 458h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 458h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 458h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.93 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT9)—Offset 45Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 45Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 45Ch

**Default:** 0000004Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.94 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT10)—Offset 460h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 460h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 460h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.95 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT10)—Offset 464h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 464h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 464h

**Default:** 0000004Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.96 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT11)—Offset 468h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 468h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 468h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.97 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT11)—Offset 46Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 46Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 46Ch

**Default:** 0000004Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.98 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT12)—Offset 470h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 470h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 470h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.99 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT12)—Offset 474h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 474h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 474h

**Default:** 0000004Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.100 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT13)—Offset 478h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 478h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 478h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.101 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT13)—Offset 47Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 47Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 47Ch

**Default:** 00000050h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x50<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.102 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT14)—Offset 480h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 480h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 480h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.103 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT14)—Offset 484h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 484h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 484h

**Default:** 00000051h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x51<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.104 Pad Configuration DW0 (PAD_CFG_DW0_DFX_PORT15)—Offset 488h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 488h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 488h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12 | 0x0<br>RW | **Pad Mode bit 2(PMODE2):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 11 | 0x0<br>RW | **Pad Mode bit 1(PMODE1):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 10 | 0x1<br>RW | **Pad Mode bit 0 (PMODE0):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.105 Pad Configuration DW1 (PAD_CFG_DW1_DFX_PORT15)—Offset 48Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 48Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 48Ch

**Default:** 00000052h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x52<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.106 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_12)—Offset 490h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 490h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 490h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.107 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_12)—Offset 494h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 494h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 494h

**Default:** 00000053h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x53<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.108 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_ALRT_N)—Offset 498h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 498h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 498h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.109 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_ALRT_N)—Offset 49Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 49Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 49Ch

**Default:** 00000054h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x54<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.110 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ5_N)—Offset 4A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4A0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4A0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.111 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ5_N)—Offset 4A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4A4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4A4h

**Default:** 00000055h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x55<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.112 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ6_N)—Offset 4A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4A8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4A8h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.113 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ6_N)—Offset 4ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4ACh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4ACh

**Default:** 00000056h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x56<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.114 Pad Configuration DW0 (PAD_CFG_DW0_PCIE_CLKREQ7_N)—Offset 4B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4B0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4B0h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.115 Pad Configuration DW1 (PAD_CFG_DW1_PCIE_CLKREQ7_N)—Offset 4B4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4B4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4B4h

**Default:** 00000057h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x57<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.116 Pad Configuration DW0 (PAD_CFG_DW0_UART0_RXD)—Offset 4B8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4B8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4B8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.117 Pad Configuration DW1 (PAD_CFG_DW1_UART0_RXD)—Offset 4BCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4BCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4BCh

**Default:** 00003058h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x58<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.118 Pad Configuration DW0 (PAD_CFG_DW0_UART0_TXD)—Offset 4C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4C0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4C0h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.119 Pad Configuration DW1 (PAD_CFG_DW1_UART0_TXD)—Offset 4C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4C4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4C4h

**Default:** 00000059h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x59<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.120 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_CLK)—Offset 4C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4C8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4C8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.121 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_CLK)—Offset 4CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4CCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4CCh

**Default:** 0000305Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.122 Pad Configuration DW0 (PAD_CFG_DW0_SMB5_GBE_DATA)—Offset 4D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4D0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4D0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.123 Pad Configuration DW1 (PAD_CFG_DW1_SMB5_GBE_DATA)—Offset 4D4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4D4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4D4h

**Default:** 0000305Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.124 Pad Configuration DW0 (PAD_CFG_DW0_ERROR2_N)—Offset 4D8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4D8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4D8h

**Default:** 45000601h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x1<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.125 Pad Configuration DW1 (PAD_CFG_DW1_ERROR2_N)—Offset 4DCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4DCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4DCh

**Default:** 0000005Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.126 Pad Configuration DW0 (PAD_CFG_DW0_ERROR1_N)—Offset 4E0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4E0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4E0h

**Default:** 45000601h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x1<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.127 Pad Configuration DW1 (PAD_CFG_DW1_ERROR1_N)—Offset 4E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4E4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4E4h

**Default:** 0000005Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.128 Pad Configuration DW0 (PAD_CFG_DW0_ERROR0_N)—Offset 4E8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4E8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4E8h

**Default:** 45000601h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x1<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.129 Pad Configuration DW1 (PAD_CFG_DW1_ERROR0_N)—Offset 4ECh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4ECh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4ECh

**Default:** 0000005Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.130 Pad Configuration DW0 (PAD_CFG_DW0_IERR_N)—Offset 4F0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4F0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4F0h

**Default:** 44000601h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x1<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.131 Pad Configuration DW1 (PAD_CFG_DW1_IERR_N)—Offset 4F4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4F4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4F4h

**Default:** 0000005Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x5f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.132 Pad Configuration DW0 (PAD_CFG_DW0_MCERR_N)—Offset 4F8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4F8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4F8h

**Default:** 44000601h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x1<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.133 Pad Configuration DW1 (PAD_CFG_DW1_MCERR_N)—Offset 4FCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 4FCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 4FCh

**Default:** 00000060h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x60<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.134 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_CLK)—Offset 500h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 500h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 500h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.135 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_CLK)—Offset 504h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 504h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 504h

**Default:** 00003061h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x61<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.136 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_DATA)—Offset 508h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 508h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 508h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.137 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_DATA)—Offset 50Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 50Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 50Ch

**Default:** 00003062h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x62<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.138 Pad Configuration DW0 (PAD_CFG_DW0_SMB0_LEG_ALRT_N)—Offset 510h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 510h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 510h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.139 Pad Configuration DW1 (PAD_CFG_DW1_SMB0_LEG_ALRT_N)—Offset 514h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 514h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 514h

**Default:** 00003063h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x63<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.140 Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_DATA)—Offset 518h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 518h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 518h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.141 Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_DATA)—Offset 51Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 51Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 51Ch

**Default:** 00003064h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x64<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.142 Pad Configuration DW0 (PAD_CFG_DW0_SMB1_HOST_CLK)—Offset 520h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 520h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 520h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.143 Pad Configuration DW1 (PAD_CFG_DW1_SMB1_HOST_CLK)—Offset 524h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 524h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 524h

**Default:** 00003065h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x65<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.144 Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_DATA)—Offset 528h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 528h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 528h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.145 Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_DATA)—Offset 52Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 52Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 52Ch

**Default:** 00003066h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x66<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.146 Pad Configuration DW0 (PAD_CFG_DW0_SMB2_PECI_CLK)—Offset 530h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 530h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 530h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.147 Pad Configuration DW1 (PAD_CFG_DW1_SMB2_PECI_CLK)—Offset 534h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 534h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 534h

**Default:** 00003067h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x67<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.148 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_DATA)—Offset 538h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 538h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 538h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.149 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_DATA)—Offset 53Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 53Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 53Ch

**Default:** 00003068h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x68<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.150 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_CLK)—Offset 540h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 540h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 540h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.151 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_CLK)—Offset 544h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 544h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 544h

**Default:** 00003069h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x69<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.152 Pad Configuration DW0 (PAD_CFG_DW0_SMB4_CSME0_ALRT_N)—Offset 548h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 548h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 548h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.153 Pad Configuration DW1 (PAD_CFG_DW1_SMB4_CSME0_ALRT_N)—Offset 54Ch

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 54Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 54Ch

**Default:** 0000306Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.154 Pad Configuration DW0 (PAD_CFG_DW0_USB_OC0_N)—Offset 550h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 550h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 550h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.155 Pad Configuration DW1 (PAD_CFG_DW1_USB_OC0_N)—Offset 554h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 554h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 554h

**Default:** 0000306Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.156 Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE0)—Offset 558h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 558h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 558h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.157 Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE0)—Offset 55Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 55Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 55Ch

**Default:** 0000006Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.158 Pad Configuration DW0 (PAD_CFG_DW0_FLEX_CLK_SE1)—Offset 560h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 560h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 560h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.159 Pad Configuration DW1 (PAD_CFG_DW1_FLEX_CLK_SE1)—Offset 564h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 564h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 564h

**Default:** 0000006Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.160 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_4)—Offset 568h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 568h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 568h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.161 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_4)—Offset 56Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 56Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 56Ch

**Default:** 0000006Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.162 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_5)—Offset 570h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 570h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 570h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.163 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_5)—Offset 574h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 574h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 574h

**Default:** 0000006Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x6f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.164 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_6)—Offset 578h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 578h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 578h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.165 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_6)—Offset 57Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 57Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 57Ch

**Default:** 00000070h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x70<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.166 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_7)—Offset 580h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 580h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 580h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.167 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_7)—Offset 584h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 584h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 584h

**Default:** 00000071h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x71<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.168 Pad Configuration DW0 (PAD_CFG_DW0_SATA0_LED_N)—Offset 588h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 588h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 588h

**Default:** 45000700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.169 Pad Configuration DW1 (PAD_CFG_DW1_SATA0_LED_N)—Offset 58Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 58Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 58Ch

**Default:** 00000072h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x72<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.170 Pad Configuration DW0 (PAD_CFG_DW0_SATA1_LED_N)—Offset 590h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 590h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 590h

**Default:** 45000700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.171 Pad Configuration DW1 (PAD_CFG_DW1_SATA1_LED_N)—Offset 594h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 594h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 594h

**Default:** 00000073h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x73<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.172 Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT0)—Offset 598h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 598h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 598h

**Default:** 45000700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.173 Pad Configuration DW1 (PAD_CFG_DW1_SATA_PDETECT0)—Offset 59Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 59Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 59Ch

**Default:** 00000074h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x74<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.174 Pad Configuration DW0 (PAD_CFG_DW0_SATA_PDETECT1)—Offset 5A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5A0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5A0h

**Default:** 45000700h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.175 Pad Configuration DW1 (PAD_CFG_DW1)—Offset 5A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5A4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5A4h

**Default:** 00000075h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x75<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.176 Pad Configuration DW0 (PAD_CFG_DW0_SATA0_SDOUT)—Offset 5A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5A8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5A8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.177 Pad Configuration DW1 (PAD_CFG_DW1_SATA0_SDOUT)—Offset 5ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5ACh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5ACh

**Default:** 00000076h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x76<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.178 Pad Configuration DW0 (PAD_CFG_DW0_SATA1_SDOUT)—Offset 5B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5B0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5B0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.179 Pad Configuration DW1 (PAD_CFG_DW1_SATA1_SDOUT)—Offset 5B4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5B4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5B4h

**Default:** 00000018h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x18<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.180 Pad Configuration DW0 (PAD_CFG_DW0_UART1_RXD)—Offset 5B8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5B8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5B8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.181 Pad Configuration DW1 (PAD_CFG_DW1_UART1_RXD)—Offset 5BCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5BCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5BCh

**Default:** 00003019h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x19<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.182 Pad Configuration DW0 (PAD_CFG_DW0_UART1_TXD)—Offset 5C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5C0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5C0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.183 Pad Configuration DW1 (PAD_CFG_DW1_UART1_TXD)—Offset 5C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5C4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5C4h

**Default:** 0000001Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.184 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_8)—Offset 5C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5C8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5C8h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.185 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_8)—Offset 5CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5CCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5CCh

**Default:** 0000001Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.186 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_9)—Offset 5D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5D0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5D0h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.187 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_9)—Offset 5D4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5D4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5D4h

**Default:** 0000001Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.188 Pad Configuration DW0 (PAD_CFG_DW0_TCK)—Offset 5D8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5D8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5D8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.189 Pad Configuration DW1 (PAD_CFG_DW1_TCK)—Offset 5DCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5DCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5DCh

**Default:** 0400101Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x4<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.190 Pad Configuration DW0 (PAD_CFG_DW0_TRST_N)—Offset 5E0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5E0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5E0h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.191 Pad Configuration DW1 (PAD_CFG_DW1_TRST_N)—Offset 5E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5E4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5E4h

**Default:** 0400301Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.192 Pad Configuration DW0 (PAD_CFG_DW0_TMS)—Offset 5E8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5E8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5E8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.193 Pad Configuration DW1 (PAD_CFG_DW1_TMS)—Offset 5ECh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5ECh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5ECh

**Default:** 0400301Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x1f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.194 Pad Configuration DW0 (PAD_CFG_DW0_TDI)—Offset 5F0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5F0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5F0h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.195 Pad Configuration DW1 (PAD_CFG_DW1_TDI)—Offset 5F4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5F4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5F4h

**Default:** 04003020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x20<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.196 Pad Configuration DW0 (PAD_CFG_DW0_TDO)—Offset 5F8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5F8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5F8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.197 Pad Configuration DW1 (PAD_CFG_DW1_TDO)—Offset 5FCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 5FCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 5FCh

**Default:** 00000021h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x21<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.198 Pad Configuration DW0 (PAD_CFG_DW0_CX_PRDY_N)—Offset 600h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 600h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 600h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.199 Pad Configuration DW1 (PAD_CFG_DW1_CX_PRDY_N)—Offset 604h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 604h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 604h

**Default:** 00003022h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x22<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.200 Pad Configuration DW0 (PAD_CFG_DW0_CX_PREQ_N)—Offset 608h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 608h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 608h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.201 Pad Configuration DW1 (PAD_CFG_DW1_CX_PREQ_N)—Offset 60Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 60Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 60Ch

**Default:** 04003023h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x23<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.202 Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGINOUT)—Offset 610h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 610h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 610h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.203 Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGINOUT)—Offset 614h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 614h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 614h

**Default:** 04003024h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x24<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.204 Pad Configuration DW0 (PAD_CFG_DW0_CTBTRIGOUT)—Offset 618h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 618h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 618h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.205 Pad Configuration DW1 (PAD_CFG_DW1_CTBTRIGOUT)—Offset 61Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 61Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 61Ch

**Default:** 04000025h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x25<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.206 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE2)—Offset 620h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 620h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 620h

**Default:** 44000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.207 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE2)—Offset 624h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 624h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 624h

**Default:** 04000026h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x26<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.208 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE3)—Offset 628h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 628h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 628h

**Default:** 44000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.209 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE3)—Offset 62Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 62Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 62Ch

**Default:** 04000027h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x27<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.210 Pad Configuration DW0 (PAD_CFG_DW0_DFX_SPARE4)—Offset 630h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 630h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 630h

**Default:** 44000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.211 Pad Configuration DW1 (PAD_CFG_DW1_DFX_SPARE4)—Offset 634h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 634h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 634h

**Default:** 00000028h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x28<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.212 Pad Configuration DW0 (PAD_CFG_DW0_SUSPWRDNACK)—Offset 638h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 638h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 638h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.213 Pad Configuration DW1 (PAD_CFG_DW1_SUSPWRDNACK)—Offset 63Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 63Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 63Ch

**Default:** 00000029h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x29<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.214 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SUSCLK)—Offset 640h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 640h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 640h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.215 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SUSCLK)—Offset 644h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 644h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 644h

**Default:** 0000002Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.216 Pad Configuration DW0 (PAD_CFG_DW0_ADR_TRIGGER)—Offset 648h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 648h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 648h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.217 Pad Configuration DW1 (PAD_CFG_DW1_ADR_TRIGGER)—Offset 64Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 64Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 64Ch

**Default:** 0000002Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.218 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S45_N)—Offset 650h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 650h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 650h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.219 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S45_N)—Offset 654h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 654h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 654h

**Default:** 0000002Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.220 Pad Configuration DW0 (PAD_CFG_DW0_PMU_SLP_S3_N)—Offset 658h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 658h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 658h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.221 Pad Configuration DW1 (PAD_CFG_DW1_PMU_SLP_S3_N)—Offset 65Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 65Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 65Ch

**Default:** 0000002Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.222 Pad Configuration DW0 (PAD_CFG_DW0_PMU_WAKE_N)—Offset 660h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 660h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 660h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.223 Pad Configuration DW1 (PAD_CFG_DW1_PMU_WAKE_N)—Offset 664h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 664h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 664h

**Default:** 0000302Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.224 Pad Configuration DW0 (PAD_CFG_DW0_PMU_PWRBTN_N)—Offset 668h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 668h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 668h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.225 Pad Configuration DW1 (PAD_CFG_DW1_PMU_PWRBTN_N)—Offset 66Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 66Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 66Ch

**Default:** 0000302Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x2f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.226 Pad Configuration DW0 (PAD_CFG_DW0_PMU_RESETBUTTON_N)—Offset 670h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 670h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 670h

**Default:** 45000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.227 Pad Configuration DW1 (PAD_CFG_DW1_PMU_RESETBUTTON_N)—Offset 674h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 674h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 674h

**Default:** 00003030h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xc<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x30<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.228 Pad Configuration DW0 (PAD_CFG_DW0_PMU_PLTRST_N)—Offset 678h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 678h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 678h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.229 Pad Configuration DW1 (PAD_CFG_DW1_PMU_PLTRST_N)—Offset 67Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 67Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 67Ch

**Default:** 00000031h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x31<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.230 Pad Configuration DW0 (PAD_CFG_DW0_SUS_STAT_N)—Offset 680h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 680h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 680h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.231 Pad Configuration DW1 (PAD_CFG_DW1_SUS_STAT_N)—Offset 684h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 684h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 684h

**Default:** 00000032h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x32<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.232 Pad Configuration DW0 (PAD_CFG_DW0_SLP_S0IX_N)—Offset 688h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 688h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 688h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.233 Pad Configuration DW1 (PAD_CFG_DW1_SLP_S0IX_N)—Offset 68Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 68Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 68Ch

**Default:** 00000033h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x33<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.234 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS0_N)—Offset 690h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 690h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 690h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.235 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS0_N)—Offset 694h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 694h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 694h

**Default:** 00003C34h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x34<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.236 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CS1_N)—Offset 698h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 698h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 698h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.237 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CS1_N)—Offset 69Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 69Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 69Ch

**Default:** 00003C35h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x35<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.238 Pad Configuration DW0 (PAD_CFG_DW0_SPI_MOSI_IO0)—Offset 6A0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6A0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6A0h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.239 Pad Configuration DW1 (PAD_CFG_DW1_SPI_MOSI_IO0)—Offset 6A4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6A4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6A4h

**Default:** 00003C36h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x36<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.240 Pad Configuration DW0 (PAD_CFG_DW0_SPI_MISO_IO1)—Offset 6A8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6A8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6A8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.241 Pad Configuration DW1 (PAD_CFG_DW1_SPI_MISO_IO1)—Offset 6ACh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6ACh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6ACh

**Default:** 00003C37h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x37<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.242 Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO2)—Offset 6B0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6B0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6B0h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.243 Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO2)—Offset 6B4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6B4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6B4h

**Default:** 00003C38h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x38<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.244 Pad Configuration DW0 (PAD_CFG_DW0_SPI_IO3)—Offset 6B8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6B8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6B8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.245 Pad Configuration DW1 (PAD_CFG_DW1_SPI_IO3)—Offset 6BCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6BCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6BCh

**Default:** 00003C39h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x39<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.246 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK)—Offset 6C0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6C0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6C0h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.247 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK)—Offset 6C4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6C4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6C4h

**Default:** 00003C3Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.248 Pad Configuration DW0 (PAD_CFG_DW0_SPI_CLK_LOOPBK)—Offset 6C8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6C8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6C8h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RO | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.249 Pad Configuration DW1 (PAD_CFG_DW1_SPI_CLK_LOOPBK)—Offset 6CCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6CCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6CCh

**Default:** 00003C3Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.250 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO0)—Offset 6D0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6D0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6D0h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.251 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO0)—Offset 6D4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6D4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6D4h

**Default:** 00003C3Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.252 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO1)—Offset 6D8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6D8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6D8h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.253 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO1)—Offset 6DCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6DCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6DCh

**Default:** 00003C3Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.254 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO2)—Offset 6E0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6E0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6E0h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.255 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO2)—Offset 6E4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6E4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6E4h

**Default:** 00003C3Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.256 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_IO3)—Offset 6E8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6E8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6E8h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.257 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_IO3)—Offset 6ECh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6ECh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6ECh

**Default:** 00003C3Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0xf<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x3f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.258 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CS0_N)—Offset 6F0h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6F0h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6F0h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.259 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CS0_N)—Offset 6F4h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6F4h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6F4h

**Default:** 00000040h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x40<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.260 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK)—Offset 6F8h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6F8h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6F8h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.261 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK)—Offset 6FCh

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 6FCh
**Host Memory Space:** SBREG_BAR + 0xC50000 + 6FCh

**Default:** 00001041h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x4<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x41<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.262 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_RST_N)—Offset 700h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 700h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 700h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.263 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_RST_N)—Offset 704h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 704h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 704h

**Default:** 00001042h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x4<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x42<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.264 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_ALRT0_N)—Offset 708h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 708h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 708h

**Default:** 45000A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.265 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_ALRT0_N)—Offset 70Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 70Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 70Ch

**Default:** 00000043h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x43<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.266 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_10)—Offset 710h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 710h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 710h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.267 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_10)—Offset 714h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 714h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 714h

**Default:** 00000044h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x44<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.268 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_11)—Offset 718h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 718h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 718h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.269 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_11)—Offset 71Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 71Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 71Ch

**Default:** 00000045h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x45<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.270 Pad Configuration DW0 (PAD_CFG_DW0_ESPI_CLK_LOOPBK)—Offset 720h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 720h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 720h

**Default:** 44000600h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x0<br>RO | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RO | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RO | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.271 Pad Configuration DW1 (PAD_CFG_DW1_ESPI_CLK_LOOPBK)—Offset 724h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 724h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 724h

**Default:** 00000046h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x46<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.272 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CMD)—Offset 728h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 728h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 728h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.273 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CMD)—Offset 72Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 72Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 72Ch

**Default:** 00000047h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x47<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.274 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_STROBE)—Offset 730h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 730h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 730h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.275 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_STROBE)—Offset 734h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 734h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 734h

**Default:** 00000048h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x48<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.276 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_CLK)—Offset 738h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 738h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 738h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.277 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_CLK)—Offset 73Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 73Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 73Ch

**Default:** 00000049h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x49<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.278 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D0)—Offset 740h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 740h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 740h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.279 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D0)—Offset 744h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 744h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 744h

**Default:** 0000004Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4a<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.280 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D1)—Offset 748h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 748h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 748h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.281 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D1)—Offset 74Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 74Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 74Ch

**Default:** 0000004Bh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4b<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.282 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D2)—Offset 750h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 750h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 750h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.283 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D2)—Offset 754h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 754h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 754h

**Default:** 0000004Ch

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4c<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.284 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D3)—Offset 758h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 758h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 758h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.285 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D3)—Offset 75Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 75Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 75Ch

**Default:** 0000004Dh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4d<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.286 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D4)—Offset 760h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 760h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 760h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.287 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D4)—Offset 764h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 764h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 764h

**Default:** 0000004Eh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4e<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.288 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D5)—Offset 768h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 768h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 768h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.289 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D5)—Offset 76Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 76Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 76Ch

**Default:** 0000004Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x4f<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.290 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D6)—Offset 770h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 770h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 770h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.291 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D6)—Offset 774h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 774h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 774h

**Default:** 00000050h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x50<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.292 Pad Configuration DW0 (PAD_CFG_DW0_EMMC_D7)—Offset 778h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 778h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 778h

**Default:** 45000300h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x1<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.293 Pad Configuration DW1 (PAD_CFG_DW1_EMMC_D7)—Offset 77Ch

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 77Ch
**Host Memory Space:** SBREG_BAR + 0xC50000 + 77Ch

**Default:** 00000051h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x51<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

## 65.3.294 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_3)—Offset 780h

**Type:** Sideband Register (Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 780h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 780h

**Default:** 45000200h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x1<br>RW | **Pad Reset Config (PADRSTCFG):** This field controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below:<br>• 00 = Powergood (i.e. sticky reset)<br>• 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b)<br>• 10 = GPIO Reset (i.e. host reset: host_rst_b)<br>• 11 = Reserved; tied to inactive (i.e. '1' for active low reset)<br>For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b<br>The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion. |
| 29 | 0x0<br>RW | **RX Pad State Select (RXPADSTSEL):** Determines from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0).<br>• 0 = Raw RX pad state directly from CFIO RX buffer<br>• 1 = Internal RX pad state (subject to RXINV and PreGfRXSel settings) |
| 28 | 0x0<br>RW | **RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV.<br>• 0 = No Override<br>• 1 = RX drive 1 internally |
| 27 | 0x0<br>RO | **Reserved (RSVD_0):** Reserved. |
| 26:25 | 0x2<br>RW | **RX Level/Edge Configuration (RXEVCFG):** Determines if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller.<br>• 0h = Level<br>• 1h = Edge<br>• 2h = Drive '0'<br>• 3h = Reserved (implement as setting 0h) |
| 24 | 0x1<br>RW | **Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determines if the synchronized version of the raw RX pad state should be subjected to glitch filter or not.<br>This field only makes sense when the RX buffer is configured as an input and is not disabled.<br>• 0 = Select synchronized, non filtered RX pad state<br>• 1 = Select synchronized, filtered RX pad state<br>The selected RX pad state can be further subjected to polarity inversion through RXINV |
| 23 | 0x0<br>RW | **RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode.<br>The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel.<br>This bit does not affect GPIORXState.<br>During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI.<br>• 0 = No inversion<br>• 1 = Inversion |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 22:21 | 0x0<br>RO | **RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode).<br>Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0.<br>• 0 = Function defined in Pad Mode controls TX and RX Enables<br>• 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally<br>• 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally<br>• 3 = Function controls TX Enabled and RX is always enabled |
| 20 | 0x0<br>RW | **GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determines if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause peripheral IRQ<br>• 1 = Routing can cause peripheral IRQ<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s). |
| 19 | 0x0<br>RW | **GPIO Input Route SCI (GPIROUTSCI):** Determines if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SCI<br>• 1 = Routing can cause SCI<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU. |
| 18 | 0x0<br>RW | **GPIO Input Route SMI (GPIROUTSMI):** Determines if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause SMI.<br>• 1 = Routing can cause SMI.<br>***Note:*** This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s). |
| 17 | 0x0<br>RW | **GPIO Input Route NMI (GPIROUTNMI):** Determines if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect.<br>• 0 = Routing does not cause NMI.<br>• 1 = Routing can cause NMI.<br>***Note:*** This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit. |
| 16:13 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved. |
| 12:10 | 0x0<br>RW | **Pad Mode bits (PMODE):** Bits [12:10] of this DW0 register determine whether the Pad is controlled by GPIO controller logic or one of the SoC native functions is assigned to the Pad.<br>Bits [12:10] decode:<br>• 0 = GPIO controls the Pad<br>• 1 = Function 1, if applicable, controls the Pad<br>• 2 = Function 2, if applicable, controls the Pad<br>• ...<br>• 7 = Function 7, if applicable, controls the Pad |
| 9 | 0x1<br>RW | **GPIO RX Disable (GPIORXDIS):**<br>• 0 = Enable the input buffer (active low enable) of the pad<br>• 1 = Disable the input buffer of the pad.<br>***Note:*** When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP. |
| 8 | 0x0<br>RW | **GPIO TX Disable (GPIOTXDIS):**<br>• 0 = Enable the output buffer (active low enable) of the pad<br>• 1 = Disable the output buffer of the pad; i.e. Hi-Z |
| 7:2 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved. |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 1 | 0x0<br>RO | **GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and is not affected by PMode and RXINV settings. |
| 0 | 0x0<br>RW | **GPIO TX State (GPIOTXSTATE):**<br>• 0 = Drive a level '0' to the TX output pad<br>• 1 = Drive a level '1' to the TX output pad |

## 65.3.295 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_3)—Offset 784h

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xC5] + 784h
**Host Memory Space:** SBREG_BAR + 0xC50000 + 784h

**Default:** 00000052h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved**. |
| 13:10 | 0x0<br>RW | **Termination (TERM):** The Pad Termination state defines the different weak Pull-Up and Pull-Down settings that are supported by the buffer. The settings for register bits [13:10] correspond to:<br>• 0h = none<br>• 2h = 5k weak Pull-Down<br>• 4h = 20k weak Pull-Down<br>• 8h = none<br>• 9h = 1k weak Pull-Up<br>• Ah = 5k weak Pull-Up<br>• Bh = 2k weak Pull-Up<br>• Ch = 20k weak Pull-Up<br>• Dh = 1k and 2k weak Pull-Up<br>• Fh = The TERM register does not define the termination. Instead, the SoC internal circuitry controls the type and value of the termination, if any, on this pad.<br>• All others bit settings are reserved. If a reserved value is programmed, the pad may malfunction.<br>The setting of this register field is applicable in all Pad Modes including GPIO. As each Pad Mode may require different termination and isolation, care must be taken by software and BIOS in the transition with appropriate register programming. The actual transition sequence requirement may vary on case-by-case basis depending on the native functions involved. For example, before changing the pad from the output to input direction, the weak Pull-Up/Pull-Down settings should be programmed first to ensure the input does not float momentarily. |
| 9:8 | 0x0<br>RO | **Reserved**. |
| 7:0 | 0x52<br>RO | **Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad.<br>• 0 = Interrupt Line 0<br>• 1 = Interrupt Line 1<br>Up to the max IOxAPIC IRQ supported by the SoC. |

§ §

# 66 SMBus - PECI

## 66.1 Introduction and Index

Much of the programming-related information and register descriptions for the SMBus-PECI are in Section 30, "SMBus Controller - Platform Environment Control Interface" on page 795.

The legend for the register access codes are described in Chapter 43, "Introduction."

§ §

# 67 Interrupt Steering and Control

## 67.1 Introduction and Index

The host-accessible registers for Interrupt Steering and Control are described here. The legend for the register access codes are described in Chapter 43, “Introduction.”

### 67.1.1 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

**Table 67-1. Summary of Sideband Registers—0xD0**

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 3100 | 80 | "PIRQA Routing Control (PARC)—Offset 3100h" on page 3741 |
| 3101 | 80 | "PIRQB Routing Control (PBRC)—Offset 3101h" on page 3742 |
| 3102 | 80 | "PIRQC Routing Control (PCRC)—Offset 3102h" on page 3743 |
| 3103 | 80 | "PIRQD Routing Control (PDRC)—Offset 3103h" on page 3744 |
| 3104 | 80 | "PIRQE Routing Control (PERC)—Offset 3104h" on page 3745 |
| 3105 | 80 | "PIRQF Routing Control (PFRC)—Offset 3105h" on page 3746 |
| 3106 | 80 | "PIRQG Routing Control (PGRC)—Offset 3106h" on page 3747 |
| 3107 | 80 | "PIRQH Routing Control (PHRC)—Offset 3107h" on page 3748 |
| 3120 | 00000000 | "Message Decoder Control (MSGDC)—Offset 3120h" on page 3749 |
| 3140 | 3210 | "PCI Interrupt Route 0 (PIR0)—Offset 3140h" on page 3750 |
| 3142 | 3210 | "PCI Interrupt Route 1 (PIR1)—Offset 3142h" on page 3751 |
| 3144 | 3210 | "PCI Interrupt Route 2 (PIR2)—Offset 3144h" on page 3752 |
| 3146 | 3210 | "PCI Interrupt Route 3 (PIR3)—Offset 3146h" on page 3753 |
| 3148 | 3210 | "PCI Interrupt Route 4 (PIR4)—Offset 3148h" on page 3754 |
| 314A | 3210 | "PCI Interrupt Route 5 (PIR5)—Offset 314Ah" on page 3755 |
| 314C | 3210 | "PCI Interrupt Route 6 (PIR6)—Offset 314Ch" on page 3756 |
| 314E | 3210 | "PCI Interrupt Route 7 (PIR7)—Offset 314Eh" on page 3757 |
| 3150 | 3210 | "PCI Interrupt Route 8 (PIR8)—Offset 3150h" on page 3758 |
| 3152 | 3210 | "PCI Interrupt Route 9 (PIR9)—Offset 3152h" on page 3759 |
| 3154 | 3210 | "PCI Interrupt Route 10 (PIR10)—Offset 3154h" on page 3760 |
| 3156 | 3210 | "PCI Interrupt Route 11 (PIR11)—Offset 3156h" on page 3761 |
| 3158 | 3210 | "PCI Interrupt Route 12 (PIR12)—Offset 3158h" on page 3762 |
| 31FC | 00000000 | "General Interrupt Control (GIC)—Offset 31FCh" on page 3763 |
| 3200 | 00FF0000 | "Interrupt Polarity Control 0 (IPC0)—Offset 3200h" on page 3765 |
| 3204 | 00000000 | "Interrupt Polarity Control 1 (IPC1)—Offset 3204h" on page 3765 |
| 3208 | 00000000 | "Interrupt Polarity Control 2 (IPC2)—Offset 3208h" on page 3765 |
| 320C | 00000000 | "Interrupt Polarity Control 3 (IPC3)—Offset 320Ch" on page 3766 |
| 3220 | 00000000 | "Interrupt Blocking Control (IBC)—Offset 3220h" on page 3766 |
| 3230 | 00000000 | "Interrupt Edge-Trigger Extension 0 (IETE0)—Offset 3230h" on page 3766 |
| 3234 | 00000000 | "Interrupt Edge-Trigger Extension 1 (IETE1)—Offset 3234h" on page 3767 |
| 3238 | 00000000 | "Interrupt Edge-Trigger Extension 2 (IETE2)—Offset 3238h" on page 3767 |
| 323C | 00000000 | "Interrupt Edge-Trigger Extension 3 (IETE3)—Offset 323Ch" on page 3767 |
| 3300 | 00000000 | "ITSS Power Reduction Control (ITSSPRC)—Offset 3300h" on page 3768 |
| 3304 | 0033 | "SIDE Clock Timing (SIDECT)—Offset 3304h" on page 3769 |
| 3306 | 0033 | "IPCI Clock Timing (IPCICT)—Offset 3306h" on page 3770 |
| 3308 | 0003 | "PGCB Clock Timing (PGCBCT)—Offset 3308h" on page 3770 |

## Table 67-1. Summary of Sideband Registers—0xD0 (Continued)

| Offset (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|
| 3320 | 00000000 | "Uncorrectable Error Mask (UEM)—Offset 3320h" on page 3771 |
| 3324 | 00000000 | "Uncorrectable Error Severity (UEV)—Offset 3324h" on page 3772 |
| 3328 | 00002000 | "Correctable Error Mask (CEM)—Offset 3328h" on page 3773 |
| 3330 | 00000000 | "NMI Control (NMI)—Offset 3330h" on page 3774 |
| 3334 | 0000 | "Master Message Control (MMC)—Offset 3334h" on page 3775 |
| 3336 | 0001 | "Master Message Status (MMSTS)—Offset 3336h" on page 3775 |
| 3400 | 00000000 | "HPET Offload Scale Value (HOFFVAL)—Offset 3400h" on page 3776 |

## 67.2 Sideband Registers

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method" on page 187.

### 67.2.1 PIRQA Routing Control (PARC)—Offset 3100h

PIRQA Routing Control Register.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3100h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3100h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQA is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQA is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQA<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

### 67.2.2 PIRQB Routing Control (PBRC)—Offset 3101h

PIRQB Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3101h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3101h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQB is routed to one of the legacy interrupts specified in bits [3:0] of this register.<br>When set, the PIRQB is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQB<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

### 67.2.3 PIRQC Routing Control (PCRC)—Offset 3102h

PIRQC Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3102h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3102h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQC is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQC is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQC<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

### 67.2.4 PIRQD Routing Control (PDRC)—Offset 3103h

PIRQD Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3103h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3103h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQD is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQD is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQD<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

## 67.2.5 PIRQE Routing Control (PERC)—Offset 3104h

PIRQE Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3104h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3104h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQE is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQE is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQE<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

### 67.2.6 PIRQF Routing Control (PFRC)—Offset 3105h

PIRQF Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3105h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3105h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQF is routed to one of the legacy interrupts specified in bits [3:0] of this register.<br>When set, the PIRQF is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQF<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

## 67.2.7 PIRQG Routing Control (PGRC)—Offset 3106h

PIRQG Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3106h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3106h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQG is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQG is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQG<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

## 67.2.8 PIRQH Routing Control (PHRC)—Offset 3107h

PIRQH Routing Control Register. It has the same programming values as PARC.

**Type:** Sideband Register
(Size: 8 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3107h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3107h

**Default:** 80h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 7 | 0x1<br>RW | **Interrupt Routing Enable (REN):**<br>When cleared, PIRQH is routed to one of the legacy interrupts specified in bits [3:0] of this register. When set, the PIRQH is not routed to the internal 8259 Programmable Interrupt Controller (PIC).<br><br>**Power Well:** Core |
| 6:4 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x0<br>RW | **IRQ Routing (IR):** IRQ destination for PIRQH<br>• 0000 - Reserved<br>• 0001 - Reserved<br>• 0010 - Reserved<br>• 0011 - IRQ3<br>• 0100 - IRQ4<br>• 0101 - IRQ5<br>• 0110 - IRQ6<br>• 0111 - IRQ7<br>• 1000 - Reserved<br>• 1001 - IRQ9<br>• 1010 - IRQ10<br>• 1011 - IRQ11<br>• 1100 - IRQ12<br>• 1101 - Reserved<br>• 1110 - IRQ14<br>• 1111 - IRQ15<br><br>**Power Well:** Core |

## 67.2.9 Message Decoder Control (MSGDC)—Offset 3120h

Message Decoder Control Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3120h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3120h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:12 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |
| 11:8 | 0x0<br>RW | **INTA-D Default Decoder PIR Mapping (INTDDPIR):** Specifies the PIR register to use for the INTA-D Default Decoder state.<br>PIR registers map INTA-D state to PRIQA-H. This value is only used when INTDDEN is set.<br>***Note:*** Acceptable values include any valid PIR register number (i.e. 0-12) or the value of 13. The value of 13, which behaves similar to the PIR direct connect source IDs, maps INTA to PIRQA, INTB maps to PIRQB, INTC maps to PIRQC and INTD maps to PIRQD and has no programmable option to change it.<br><br>**Power Well:** Core |
| 7:3 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **NMI Default Decoder Enable (NMIDDEN):**<br>When set, the NMI decode logic will accept ASSERT_NMI/DEASSERT_NMI messages from unknown sources (sources not specified with the NMI_SRC_ID and (if OPC_ASRT/DSRTNMI == OPC_ASRT/DASRTIOCHK which is the default case) IOCHK_SRC_ID parameters).<br>When cleared, NMI messages from unknown sources will be dropped.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **INTAD Default Decoder Enable (INTDDEN):**<br>When set, the INTA-D decoder logic will accept ASSERT_INTA-D/DEASSERT_INTA-D messages from unknown sources (source not specified with the IPIR0-12_SRC_ID and PIRDC_SRC_ID parameters).<br>When cleared, INTA-D messages from unknown sources will be dropped.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **IRQN Default Decoder Enable (IRQDDEN):**<br>When set, the IRQN decoder logic will accept ASSERT_IRQN/DEASSERT_IRQN messages from unknown source (source not specified with the SIRQ_SINGLE_CAUSE_SRC_ID and SIRQ_MULTI_CAUSE_SRC_ID parameters).<br>When cleared, IRQN messages from unknown sources will be dropped.<br><br>**Power Well:** Core |

## 67.2.10 PCI Interrupt Route 0 (PIR0)—Offset 3140h

PCI Interrupt Route 0 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3140h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3140h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.11 PCI Interrupt Route 1 (PIR1)—Offset 3142h

PCI Interrupt Route 1 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3142h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3142h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.12 PCI Interrupt Route 2 (PIR2)—Offset 3144h

PCI Interrupt Route 2 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3144h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3144h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.13 PCI Interrupt Route 3 (PIR3)—Offset 3146h

PCI Interrupt Route 3 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3146h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3146h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.14 PCI Interrupt Route 4 (PIR4)—Offset 3148h

PCI Interrupt Route 4 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3148h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3148h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.15 PCI Interrupt Route 5 (PIR5)—Offset 314Ah

PCI Interrupt Route 5 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 314Ah
**Host Memory Space:** SBREG_BAR + 0xD00000 + 314Ah

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.16 PCI Interrupt Route 6 (PIR6)—Offset 314Ch

PCI Interrupt Route 6 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 314Ch
**Host Memory Space:** SBREG_BAR + 0xD00000 + 314Ch

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.17 PCI Interrupt Route 7 (PIR7)—Offset 314Eh

PCI Interrupt Route 7 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 314Eh
**Host Memory Space:** SBREG_BAR + 0xD00000 + 314Eh

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.18 PCI Interrupt Route 8 (PIR8)—Offset 3150h

PCI Interrupt Route 8 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3150h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3150h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.19 PCI Interrupt Route 9 (PIR9)—Offset 3152h

PCI Interrupt Route 9 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3152h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3152h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.20 PCI Interrupt Route 10 (PIR10)—Offset 3154h

PCI Interrupt Route 10 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3154h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3154h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.21 PCI Interrupt Route 11 (PIR11)—Offset 3156h

PCI Interrupt Route 11 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3156h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3156h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.22 PCI Interrupt Route 12 (PIR12)—Offset 3158h

PCI Interrupt Route 12 Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3158h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3158h

**Default:** 3210h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 14:12 | 0x3<br>RW | **Interrupt D Pin Route (IDR):** Indicates which PIRQ internal to the SoC is connected to the INTD#.<br>• 0h - PIRQA#<br>• 1h - PIRQB#<br>• 2h - PIRQC#<br>• 3h - PIRQD#<br>• 4h - PIRQE#<br>• 5h - PIRQF#<br>• 6h - PIRQG#<br>• 7h - PIRQH#<br><br>**Power Well:** Core |
| 11 | 0x0<br>RO | **Reserved (RSVD_1):** Reserved<br><br>**Power Well:** Core |
| 10:8 | 0x2<br>RW | **Interrupt C Pin Route (ICR):** See the IDR description. This field applies to INTC#.<br><br>**Power Well:** Core |
| 7 | 0x0<br>RO | **Reserved (RSVD_2):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x1<br>RW | **Interrupt B Pin Route (IBR):** See the IDR description. This field applies to INTB#.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD_3):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x0<br>RW | **Interrupt A Pin Route (IAR):** See the IDR description. This field applies to INTA#.<br><br>**Power Well:** Core |

## 67.2.23 General Interrupt Control (GIC)—Offset 31FCh

FEC10000h - FEC3FFFFh is allocated to PCIe when Port I/OxAPIC Enable (PAE) bit is set.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 31FCh
**Host Memory Space:** SBREG_BAR + 0xD00000 + 31FCh

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 21 | 0x0<br>RW | **CF9 Completion Dependency Disable (CF9NODEP):** When set, the completion ordering between the Non-Posted IO Write causing CF9 Request to PMC and the GO_S1_SENT indication is disabled.<br>In other words, the IO Write completion can be issued before PMC indicates the GO_S1_SENT indication.<br>When cleared, the completion for the IO Write is only sent after the GO_S1_SENT indication asserts.<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW | **INIT# VLW Message Completion Dependacy Disable (INITVMNODEP):** When sent, the completion ordering between the Non-Posted IO Write which causes an INIT and the Non-Posted VLW message which signals the INIT is broken.<br>In other words, the IO Write completion can be sent out before the VLW completion is received.<br>When cleared, the completion for the IO Write is only be sent AFTER the completion for the VLW is received.<br><br>**Power Well:** Core |
| 19 | 0x0<br>RW | **INIT# VLW Message Disable (INITVMDIS):** Disable INIT# VLW message.<br>When set, if other type of VLW message to be sent, the VLWValue Bit[5] under VLW Vendor Defined Message should always be 0 - no active edge on INIT#.<br>When clear, the INIT# information will be conveyed as normal.<br><br>**Power Well:** Core |
| 18 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 17 | 0x0<br>RW | **Alternate Access Mode Enable (AME):** When set, read only registers can be written, and write only registers can be read.<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW | **Shutdown Policy Select (SDPS):** When cleared (default) the PCH will update INIT# in response to the shutdown Vendor Defined Message (VDM).<br>When set to 1, PCH will treat the shutdown VDM similar to receiving a CF9h I/O write, and will drive PLTRST# active.<br>See Section 16.13 Reset Behavior for variations on whether power is cycled due to this reset source depending on the programming of other bits.<br>BIOS guide note: This register is reset any time PLTRST# asserts.<br><br>**Power Well:** Core |
| 15:9 | 0x0<br>RW | **MAX_IRQ_ENTRY_SIZE (MAXIRQSIZE):** This field indicates the size of the IOxAPIC entry.<br>The default size is 120 entries.<br>• 0000000: 120 entry size<br>• 0000001: 24 entry size (Legacy mode)<br>• 0000010 - 1111111: Reserved<br><br>**Power Well:** Core |

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 8 | 0x0<br>RW | **Server Error Reporting Mode (SERM):** When set, the CPU Complex is the final target of all host space errors.<br>In this mode, if the PCH detects a fatal, non-fatal, or correctable error on DMI or PSFs, it sends one of ERR_FATAL, ERR_NONFATAL, or ERR_CORR to CPU Complex.<br>When cleared, the SoC is the final target of all host space errors.<br><br>**Power Well:** Core |
| 7:2 | 0x0<br>RW | **Reserved (RSVD2):** Scratch pad: These bits are reserved for future use.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RW | **INIT# VLW Message Disable Mode (INITVMDISMODE):** This policy bit dictates the INIT# handling when INITVMDIS is configured to 1.<br>When set to 1 and INITVMDIS = 1, the INIT# events will be dropped such that upon clearing the INITVMDIS = 0 later, no INIT# VLW Message is released.<br>When set to 0 (default) and INITVMDIS = 1, the INIT# events will be masked such that upon clearing the INITVMDIS = 0 later, the INIT# VLW Message is released.<br>**Power Well:** Core |
| 0 | 0x0<br>RO/P | **CPU Shutdown Status (CPUSDSTS):** This bit is set to 1 if the CPU sends the Shutdown Special cycle message.<br>The Shutdown Message is recognized as an INIT# event if the Shutdown Policy Select = 0.<br>Otherwise the SoC treats the Shutdown Special cycle as a request for CF9 Hard Reset.<br>This is a sticky Read Only bit that is only reset by a loss of core power.<br><br>**Power Well:** Core |

## 67.2.24 Interrupt Polarity Control 0 (IPC0)—Offset 3200h

Interrupt Polarity Control 0 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3200h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3200h

**Default:** 00FF0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0xff0000<br>RW | **IRQ 31-0 Active High Polarity Disable (IPC0_IRQxAHPOLDIS):** IRQ 31-0 Active High Polarity Disable (IPC0.IRQxAHPOLDIS):<br>When set to 1, the interrupt polarity associated with IRQ31 down to IRQ0 is inverted to appear as Active Low to IOxAPIC.<br>When set to 0, the interrupt is appears as Active High to IOxAPIC.<br><br>**Power Well:** Core |

## 67.2.25 Interrupt Polarity Control 1 (IPC1)—Offset 3204h

Interrupt Polarity Control 1 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3204h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3204h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 63-32 Active High Polarity Disable (IPC1_IRQAHPOLDIS):** IRQ 63-32 Active High Polarity Disable (IPC1.IRQAHPOLDIS):<br>When set to 1, the interrupt polarity associated with IRQ63 down to IRQ32 is inverted to appear as Active Low to IOxAPIC.<br>When set to 0, the interrupt is appears as Active High to IOxAPIC.<br><br>**Power Well:** Core |

## 67.2.26 Interrupt Polarity Control 2 (IPC2)—Offset 3208h

Interrupt Polarity Control 2 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3208h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3208h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 95-64 Active High Polarity Disable (IPC2_IRQAHPOLDIS):** IRQ 95-64 Active High Polarity Disable (IPC2.IRQAHPOLDIS):<br>When set to 1, the interrupt polarity associated with IRQ95 down to IRQ64 is inverted to appear as Active Low to IOxAPIC.<br>When set to 0, the interrupt is appears as Active High to IOxAPIC.<br><br>**Power Well:** Core |

## 67.2.27 Interrupt Polarity Control 3 (IPC3)—Offset 320Ch

Interrupt Polarity Control 3 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 320Ch
**Host Memory Space:** SBREG_BAR + 0xD00000 + 320Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 23:0 | 0x0<br>RW | **IRQ 119-96 Active High Polarity Disable (IPC3_IRQAHPOLDIS):** IRQ 119-96 Active High Polarity Disable (IPC3.IRQAHPOLDIS):<br>When set to 1, the interrupt polarity associated with IRQ119 down to IRQ96 is inverted to appear as Active Low to IOxAPIC.<br>When set to 0, the interrupt is appears as Active High to IOxAPIC.<br><br>**Power Well:** Core |

## 67.2.28 Interrupt Blocking Control (IBC)—Offset 3220h

Interrupt Blocking Control Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3220h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3220h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 23-0 IRQx Interrupt Disable (IRQxDIS):**<br>When set to 1, any interrupts associated with IRQ23 down to IRQ0 originating from ASSERT_IRQn sideband messages are blocked from reaching the 8259 and IOxAPIC interrupt controllers.<br>When set to 0, interrupts are allowed to pass to the interrupt controllers.<br>SIRQ_NUM_OF_SHAREABLE parameter value will affect the bits function.bit SIRQ_NUM_OF_SHAREABLE-1:0] will act as IRQxDIS function, bit [31:SIRQ_NUM_OF_SHAREABLE] will be reserved.<br><br>**Power Well:** Core |

## 67.2.29 Interrupt Edge-Trigger Extension 0 (IETE0)—Offset 3230h

Interrupt Edge-Trigger Extension 0 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3230h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3230h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 31-0 Disable Edge-Triggered Interrupt Extension (IETE31_0):** When set, DSn bits corresponding to Edge-triggered interrupts gets cleared if the interrupt source is de-asserted before interrupt gets serviced (this is backward compatible with previous architectures).<br>When cleared, the IOxAPIC DSn bit corresponding to Edge-triggered interrupts stays asserted (i.e. extended) even when interrupt source is de-asserted before interrupt gets serviced.<br><br>**Power Well:** Core |

## 67.2.30 Interrupt Edge-Trigger Extension 1 (IETE1)—Offset 3234h

Interrupt Edge-Trigger Extension 1 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3234h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3234h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 63-32 Disable Edge-Triggered Interrupt Extension (IETE63_32):** When set, DSn bits corresponding to Edge-triggered interrupts gets cleared if the interrupt source is de-asserted before interrupt gets serviced (this is backward compatible with previous architectures).<br>When cleared, the IOxAPIC DSn bit corresponding to Edge-triggered interrupts stays asserted (i.e. extended) even when interrupt source is de-asserted before interrupt gets serviced.<br><br>**Power Well:** Core |

## 67.2.31 Interrupt Edge-Trigger Extension 2 (IETE2)—Offset 3238h

Interrupt Edge-Trigger Extension 2 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3238h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3238h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:0 | 0x0<br>RW | **IRQ 95-64 Disable Edge-Triggered Interrupt Extension (IETE95_64):** When set, DSn bits corresponding to Edge-triggered interrupts gets cleared if the interrupt source is de-asserted before interrupt gets serviced (this is backward compatible with previous architectures).<br>When cleared the IOxAPIC DSn bit corresponding to Edge-triggered interrupts stays asserted (i.e. extended) even when interrupt source is de-asserted before interrupt gets serviced.<br><br>**Power Well:** Core |

## 67.2.32 Interrupt Edge-Trigger Extension 3 (IETE3)—Offset 323Ch

Interrupt Edge-Trigger Extension 3 Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 323Ch
**Host Memory Space:** SBREG_BAR + 0xD00000 + 323Ch

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:24 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |
| 23:0 | 0x0<br>RW | **IRQ 119-96 Disable Edge-Triggered Interrupt Extension (IETE119_96):** When set, DSn bits corresponding to Edge-triggered interrupts gets cleared if the interrupt source is de-asserted before interrupt gets serviced (this is backward compatible with previous architectures).<br>When cleared the IOxAPIC DSn bit corresponding to Edge-triggered interrupts stays asserted (i.e. extended) even when interrupt source is de-asserted before interrupt gets serviced.<br><br>**Power Well:** Core |

## 67.2.33 ITSS Power Reduction Control (ITSSPRC)—Offset 3300h

Power controls for the entire interrupt and timer subsystem.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3300h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3300h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:5 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 4 | 0x0<br>RW | **PGCB Dynamic Clock Gating Enable (PGCBDCGE):** When set, the ITSS enables dynamic clock gating of the PGCB clock domain.<br>BIOS is requested to program this field to 1.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RW | **HPET Dynamic Clock Gating Enable (HPETDCGE):** When set, the HPET enables dynamic clock gating.<br>BIOS is requested to program this field to 1.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **8254 Static Clock Gating Enable (CGE8254):** When set, the 8254 Programmable Interval Timer is disabled statically.<br>This bit shall be set by BIOS if the 8254 feature is not needed in the system or before BIOS hands off the system that supports C11. Normal operation of 8254 requires this bit to 0.<br>**Power Well:** Core |
| 1 | 0x0<br>RW | **Sideband Dynamic Clock Gating Enable (SBDCGE):** Setting this bit will enable all dynamic clock gating of the Sideband Clock domain.<br>BIOS is requested to program this field to 1.<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW | **PCI Dynamic Clock Gating Enable (PCIDCGE):** Setting this bit will enable dynamic clock gating for the ITSS Core Logic that uses PCI Clock.<br>BIOS is requested to program this field to 1.<br>**Power Well:** Core |

## 67.2.34 SIDE Clock Timing (SIDECT)—Offset 3304h

SIDE Clock Timing Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3304h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3304h

**Default:** 0033h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:7 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x3<br>RW | **Clock Request Hold-Off (CLKREQHO):** Defines the amount of idle time required between locking for power gate preparation and deassertion of the prim_clkreq signal.<br>This field defines the exponent such that the actual delay = 2^ CLKREQHO.<br>**WARNING:** This value should not be updated unless the Sideband Interface Clock Gating Enable (ITSSPRC.SBDCGE) control register bit is set to 0.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x3<br>RW | **Clock Gating Hold-Off (CLKGATEHO):** Defines the amount of idle time required before local clock gating will be engaged (if enabled).<br>This field defines the exponent such that the actual delay = 2^ CLGATEHO.<br>**WARNING:** This value should not be updated unless the Sideband Interface Clock Gating Enable (ITSSPRC.SBDCGE) control register bit is set to 0.<br><br>**Power Well:** Core |

## 67.2.35 IPCI Clock Timing (IPCICT)—Offset 3306h

IPCI Clock Timing Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3306h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3306h

**Default:** 0033h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:7 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |
| 6:4 | 0x3<br>RW | **Clock Request Hold-Off (CLKREQHO):** Defines the amount of idle time required between locking for power gate preparation and deassertion of the prim_clkreq signal.<br>This field defines the exponent such that the actual delay = 2^ CLKREQHO.<br>**WARNING:** This value should not be updated unless the PCI Dynamic Clock Gate Enable (ITSSPRC.PCIDCGE) control register bit is set to 0.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 2:0 | 0x3<br>RW | **Clock Gating Hold-Off (CLKGATEHO):** Defines the amount of idle time required before local clock gating will be engaged (if enabled). This field defines the exponent such that the actual delay = 2^ CLGATEHO.<br>**WARNING:** This value should not be updated unless the PCI Dynamic Clock Gate Enable (ITSSPRC.PCIDCGE) control register bit is set to 0.<br><br>**Power Well:** Core |

## 67.2.36 PGCB Clock Timing (PGCBCT)—Offset 3308h

PGCB Clock Timing Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3308h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3308h

**Default:** 0003h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:4 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |
| 3:0 | 0x3<br>RW | **Clock Gating Hold-Off (CLKGATEHO):** Specifies the minimum number of delay clocks the PGCB clock gate sequencing should wait, before enabling the PGCB clock to be gated at the trunk-level.<br>**WARNING:** This value should not be updated unless the PGCB Dynamic Clock Gate Enable (ITSSPRC.PGCBDCGE) control register bit is set to 0.<br><br>**Power Well:** Core |

## 67.2.37 Uncorrectable Error Mask (UEM)—Offset 3320h

When set, the corresponding error is masked, and the reported error will cause no action. When cleared, the corresponding error is enabled. This register is only reset by a loss of core power.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3320h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3320h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **Reserved (RSVD4):** Reserved<br><br>**Power Well:** Core |
| 22 | 0x0<br>RW/P | **Uncorrectable Internal Error Mask (UIE):** Mask for uncorrectable internal errors (to be used for parity errors).<br><br>**Power Well:** Core |
| 21 | 0x0<br>RO | **Reserved (RSVD3):** Reserved<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW/P | **Unsupported Request Error Mask (URE):** Mask for uncorrectable errors<br><br>**Power Well:** Core |
| 19:17 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW/P | **Unexpected Completion Mask (UC):** Mask for unexpected completions.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |

## 67.2.38 Uncorrectable Error Severity (UEV)—Offset 3324h

This register gives the option to make an uncorrectable error fatal or non-fatal. An error is fatal if the bit is set. An error is non-fatal is the bit is cleared. This register is only reset by a loss of core power.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3324h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3324h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:23 | 0x0<br>RO | **Reserved (RSVD4):** Reserved<br><br>**Power Well:** Core |
| 22 | 0x0<br>RW/P | **Uncorrectable Internal Error Severity (UIE):** Severity for uncorrectable internal errors (to be used for parity errors).<br><br>**Power Well:** Core |
| 21 | 0x0<br>RO | **Reserved (RSVD3):** Reserved<br><br>**Power Well:** Core |
| 20 | 0x0<br>RW/P | **Unsupported Request Error Severity (URE):** Severity for uncorrectable errors<br><br>**Power Well:** Core |
| 19:17 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 16 | 0x0<br>RW/P | **Unexpected Completion Severity (UC):** Severity for unexpected completions.<br><br>**Power Well:** Core |
| 15:0 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |

## 67.2.39 Correctable Error Mask (CEM)—Offset 3328h

When set, the corresponding error is masked and the reported error will cause no action. When cleared, the corresponding error is enabled. This register is only reset by a loss of core power.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3328h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3328h

**Default:** 00002000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved (RSVD2):** Reserved<br><br>**Power Well:** Core |
| 13 | 0x1<br>RW/P | **Advisory Non-Fatal Error Mask (ANFEM):** When set, masks Advisory Non-Fatal errors from signaling ERR_COR to the device control register.<br>Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.<br><br>**Power Well:** Core |
| 12:0 | 0x0<br>RO | **Reserved (RSVD1):** Reserved<br><br>**Power Well:** Core |

## 67.2.40 NMI Control (NMI)—Offset 3330h

NMI Control Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3330h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3330h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:6 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 5 | 0x0<br>RW/1C | **NMI VLW Delivery Status (NMI_VLW_STS):** A writable status bit which is set whenever a VLW message that assert NMI is successfully sent and the associated completion is received.<br>Its intent is to enable a pooling mechanism for software to enforce NMI_NOW message ordering.<br>Once set, writing the bit to1 will clear the status. Writing the bit to 0 has no effect.<br><br>**Power Well:** Core |
| 4 | 0x0<br>RO/V | **NMI Status (NMI_STS):** RO status bit indicating the current NMI status.<br>The bit will be set to (1b1) if any NMI source is asserted and NMI2SMI_EN is set to (1b0).<br>***Note:*** In models *not* compiled with the USE_NMI_CTRL_REG, the value of NMI2SMI_EN will come from the mu1_smnmi2smi_en interface pin rather than the NMI2SMI_EN control register bit.<br><br>**Power Well:** Core |
| 3 | 0x0<br>RO/V | **NMI-to-SMI Status (NMI2SMI_STS):** RO status bit indicating the current NMI2SMI status. The bit will be set (1b1) if any NMI source is asserted and NMI2SMI_EN is set (1b1).<br>***Note:*** In models *not* compiled with the USE_NMI_CTRL_REG, the value of NMI2SMI_EN will come from the mu1_smnmi2smi_en interface pin rather than the NMI2SMI_EN control register bit.<br><br>**Power Well:** Core |
| 2 | 0x0<br>RW | **NMI-to-SMI Enable (NMI2SMI_EN):** Setting to 1b1 causes NMIs to be sent as ASSERT_SMI/ DEASSERT_SMI messages to PMC instead of the regular NMI messages. Setting to 1b0 maintains the regular NMI routing (as VLW and VM). Bits NMI_NOW and NMI2SMI_EN can't be configured on same cycle.<br><br>**Power Well:** Core |
| 1 | 0x0<br>RO/V | **NMI NOW Status (NMI_NOW_STS):** RO status bit indicating the current state of NMI_NOW.<br>See NMI_NOW.<br><br>**Power Well:** Core |
| 0 | 0x0<br>WO | **NMI Now Command (NMI_NOW):** Writing 1b1 to NMI_NOW inverts the NMI NOW Status (NMI_NOW_STS) value.<br>The first time NMI_NOW is written sets the NMI_NOW_STS and initiates an NMI.<br>The next write clears the NMI_NOW_STS and allows initiating NMI by the next write to NMI_NOW.<br>Writing 1b0 to NMI_NOW has no effect. Bits NMI_NOW and NMI2SMI_EN can't be configured on same cycle.<br><br>**Power Well:** Core |

## 67.2.41 Master Message Control (MMC)—Offset 3334h

Master Message Control Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3334h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3334h

**Default:** 0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 0 | 0x0<br>RW/V | **Master Message Enable (MSTRMSG_EN):**<br>When set, allows ITSS to release any pending/in progress IOAPIC MWr, HPET MWr, VLW_EVT or the ERR Messages to the IOSF SB.<br>When cleared, ITSS prevents these messages from being issued to the IOSF SB.<br>The Master Message Enable sets to 1 when:<br>• BIOS SW write to 1<br>• ITSS receives the IOSF SB MSTR_MSG_EN Message.<br>The Master Message Enable clears to 0 when:<br>• BIOS SW write to 0<br>• ITSS receives the IOSF SB MSTR_MSG_DIS Message<br>• ITSS receives the RESET_PREP Message<br>• ITSS receives the FORCEPWRGOK Message<br>***Note:*** This bit supersedes the INITVMDIS policy bit. For example, if MSTRMSGEN = 0 but INITVMDIS = 0, INIT# VLW message is disallowed.<br><br>**Power Well:** Core |

## 67.2.42 Master Message Status (MMSTS)—Offset 3336h

Master Message Status Register

**Type:** Sideband Register
(Size: 16 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3336h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3336h

**Default:** 0001h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 15:1 | 0x0<br>RO | **Reserved (RSVD):** Reserved Field<br><br>**Power Well:** Core |
| 0 | 0x1<br>RO/V | **Master Message Locked (MSTRMSGLOC_STS):**<br>Sets when the Master Message Enable (MSTRMSG_EN) bit is set to 0 and all Sideband Message Mastering has stopped.<br>Clears when MSTRMSG_EN==1.<br>This bit has no direct effect on hardware and is used as a semaphore bit to indicate Message Mastering on IOSF SB is busy/not complete.<br><br>**Power Well:** Core |

## 67.2.43 HPET Offload Scale Value (HOFFVAL)—Offset 3400h

HPET Offload Scale Value Register

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** [Port: 0xD0] + 3400h
**Host Memory Space:** SBREG_BAR + 0xD00000 + 3400h

**Default:** 00000000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:28 | 0x0<br>RO | **Reserved (RSVD):** Reserved<br><br>**Power Well:** Core |
| 27:0 | 0x0<br>RW/P | **Offload Scale Value (VAL):** Offload Scale Value (VAL): Indicates the lower 28 bits of the value that should be added to the slow clock counter during RTC offload of the HPET timers.<br><br>**Power Well:** Core |

§ §

# 68 PHY - High-Speed I/O

## 68.1 Introduction and Index

The host-accessible registers for the PHY circuitry for the High-Speed I/O (HSIO) interface. Only the registers associated with customer PHY Tuning are shown. The HSIO interface is comprised of the:

- PCI Express* HSIO Lanes
- SATA HSIO Lanes
- USB 3.0 HSIO Lanes

The SoC has 20 HSIO Lanes maximum and are configured as shown in the Chapter 10, "Flexible I/O Adapter (FIA) Overview."

To determine the number of HSIO lanes and interface types available refer to the following:

- Table 1-1, "Features by product SKU - Server and Cloud Storage SKUs 0 through 4"
- Table 1-2, "Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10"
- Table 1-3, "Features by Product SKU - Extended Temperature SKUs 11 through 14".

The legend for the register access codes are described in Chapter 43, "Introduction."

Please refer to the document #568853 for the detail about how to tune the registers for HSIO interfaces.

### 68.1.1 Sideband Registers

An index of the Sideband registers that are accessible to the Host are listed here. Each is accessible in Host Memory Space using the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method".

**Table 68-1. Summary of Sideband Registers (Lane 0 of 19) — 0xBA**

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 28 | BA_0028 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 94 | BA_0094 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 98 | BA_0098 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 9C | BA_009C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| CC | BA_00CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 150 | BA_0150 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 154 | BA_0154 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 164 | BA_0164 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 168 | BA_0168 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

**Table 68-2. Summary of Sideband Registers (Lane 1 of 19) — 0xBA**

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 228 | BA_0228 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 294 | BA_0294 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 298 | BA_0298 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 29C | BA_029C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 2CC | BA_02CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 350 | BA_0350 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 354 | BA_0354 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 364 | BA_0364 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 368 | BA_0368 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-3. Summary of Sideband Registers (Lane 2 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 428 | BA_0428 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 494 | BA_0494 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 498 | BA_0498 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 49C | BA_049C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 4CC | BA_04CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 550 | BA_0550 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 554 | BA_0554 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 564 | BA_0564 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 568 | BA_0568 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-4. Summary of Sideband Registers (Lane 3 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 628 | BA_0628 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 694 | BA_0694 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 698 | BA_0698 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 69C | BA_069C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 6CC | BA_06CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 750 | BA_0750 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 754 | BA_0754 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 764 | BA_0764 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 768 | BA_0768 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-5. Summary of Sideband Registers (Lane 4 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 828 | BA_0828 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 894 | BA_0894 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 898 | BA_0898 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 89C | BA_089C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 8CC | BA_08CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 950 | BA_0950 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 954 | BA_0954 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 964 | BA_0964 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 968 | BA_0968 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-6. Summary of Sideband Registers (Lane 5 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| A28 | BA_0A28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| A94 | BA_0A94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| A98 | BA_0A98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| A9C | BA_0A9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| ACC | BA_0ACC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| B50 | BA_0B50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| B54 | BA_0B54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| B64 | BA_0B64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| B68 | BA_0B68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-7. Summary of Sideband Registers (Lane 6 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| C28 | BA_0C28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| C94 | BA_0C94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| C98 | BA_0C98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| C9C | BA_0C9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| CCC | BA_0CCC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| D50 | BA_0D50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| D54 | BA_0D54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| D64 | BA_0D64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| D68 | BA_0D68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-8. Summary of Sideband Registers (Lane 7 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| E28 | BA_0E28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| E94 | BA_0E94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| E98 | BA_0E98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| E9C | BA_0E9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| ECC | BA_0ECC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| F50 | BA_0F50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| F54 | BA_0F54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| F64 | BA_0F64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| F68 | BA_0F68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-9. Summary of Sideband Registers (Lane 8 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1028 | BA_1028 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1094 | BA_1094 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1098 | BA_1098 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 109C | BA_109C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 10CC | BA_10CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1150 | BA_1150 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1154 | BA_1154 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1164 | BA_1164 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1168 | BA_1168 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-10. Summary of Sideband Registers (Lane 9 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1228 | BA_1228 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1294 | BA_1294 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1298 | BA_1298 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 129C | BA_129C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 12CC | BA_12CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1350 | BA_1350 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1354 | BA_1354 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1364 | BA_1364 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1368 | BA_1368 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-11. Summary of Sideband Registers (Lane 10 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1428 | BA_1428 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1494 | BA_1494 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1498 | BA_1498 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 149C | BA_149C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 14CC | BA_14CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1550 | BA_1550 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1554 | BA_1554 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1564 | BA_1564 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1568 | BA_1568 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-12. Summary of Sideband Registers (Lane 11 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1628 | BA_1628 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1694 | BA_1694 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1698 | BA_1698 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 169C | BA_169C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 16CC | BA_16CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1750 | BA_1750 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1754 | BA_1754 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1764 | BA_1764 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1768 | BA_1768 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-13. Summary of Sideband Registers (Lane 12 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1828 | BA_1828 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1894 | BA_1894 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1898 | BA_1898 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 189C | BA_189C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 18CC | BA_18CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1950 | BA_1950 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1954 | BA_1954 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1964 | BA_1964 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1968 | BA_1968 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-14. Summary of Sideband Registers (Lane 13 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1A28 | BA_1A28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1A94 | BA_1A94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1A98 | BA_1A98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 1A9C | BA_1A9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 1ACC | BA_1ACC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1B50 | BA_1B50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1B54 | BA_1B54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1B64 | BA_1B64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1B68 | BA_1B68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-15. Summary of Sideband Registers (Lane 14 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1C28 | BA_1C28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1C94 | BA_1C94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1C98 | BA_1C98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 1C9C | BA_1C9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 1CCC | BA_1CCC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1D50 | BA_1D50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1D54 | BA_1D54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1D64 | BA_1D64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1D68 | BA_1D68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-16. Summary of Sideband Registers (Lane 15 of 19) — 0xBA

| Offset from Sideband Port BAh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 1E28 | BA_1E28 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 1E94 | BA_1E94 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 1E98 | BA_1E98 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 1E9C | BA_1E9C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 1ECC | BA_1ECC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 1F50 | BA_1F50 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 1F54 | BA_1F54 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 1F64 | BA_1F64 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 1F68 | BA_1F68 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-17. Summary of Sideband Registers (Lane 16 of 19) — 0xBB

| Offset from Sideband Port BBh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2028 | BB_2028 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 2094 | BB_2094 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 2098 | BB_2098 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 209C | BB_209C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 20CC | BB_20CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 2150 | BB_2150 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 2154 | BB_2154 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 2164 | BB_2164 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 2168 | BB_2168 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

### Table 68-18. Summary of Sideband Registers (Lane 17 of 19) — 0xBB

| Offset from Sideband Port BBh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2228 | BB_2228 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 2294 | BB_2294 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 2298 | BB_2298 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 229C | BB_229C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 22CC | BB_22CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 2350 | BB_2350 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 2354 | BB_2354 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 2364 | BB_2364 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 2368 | BB_2368 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-19. Summary of Sideband Registers (Lane 18 of 19) — 0xBB

| Offset from Sideband Port BBh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2428 | BB_2428 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 2494 | BB_2494 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 2498 | BB_2498 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 249C | BB_249C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 24CC | BB_24CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 2550 | BB_2550 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 2554 | BB_2554 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 2564 | BB_2564 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 2568 | BB_2568 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## Table 68-20. Summary of Sideband Registers (Lane 19 of 19) — 0xBB

| Offset from Sideband Port BBh (hex) | Memory-Mapped I/O offset from SBREG_BAR (hex) | Default Value (hex) | Register Name (Register Symbol) |
|---|---|---|---|
| 2628 | BB_2628 | 01020180 | "PCS_DWORD10 (pcs_dword10)—Offset 28h" on page 3788 |
| 2694 | BB_2694 | 28292A00 | "TX_DWORD5 (tx_dword5)—Offset 94h" on page 3789 |
| 2698 | BB_2698 | 151F2020 | "TX_DWORD6 (tx_dword6)—Offset 98h" on page 3790 |
| 269C | BB_269C | 022A0000 | "TX_DWORD7 (tx_dword7)—Offset 9Ch" on page 3791 |
| 26CC | BB_26CC | 201F0000 | "TX_DWORD19 (tx_dword19)—Offset CCh" on page 3792 |
| 2750 | BB_2750 | 003F3F3F | "RX_DWORD20 (rx_dword20)—Offset 150h" on page 3792 |
| 2754 | BB_2754 | 80C00000 | "RX_DWORD21 (rx_dword21)—Offset 154h" on page 3793 |
| 2764 | BB_2764 | 00005400 | "RX_DWORD25 (rx_dword25)—Offset 164h" on page 3794 |
| 2768 | BB_2768 | B800822A | "RX_DWORD26 (rx_dword26)—Offset 168h" on page 3794 |

## 68.2 Sideband Registers

The Sideband registers accessible to the Host are described here. They are accessible in Host Memory Space starting at the Sideband Base Address Register (SBREG_BAR). See Section 3.4.1, "Using SBREG_BAR Method".

### 68.2.1 PCS_DWORD10 (pcs_dword10)—Offset 28h

PCS_DWORD10 for HSIO Lane 0 is shown here. This register description applies to the PCS_DWORD10 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 28h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 28h

**Default:** 01020180

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:7 | 0x0<br>RO | **Reserved** |
| 6:5 | 0x0<br>RW | **reg_cfg_data_dynclkgate_mode_1_0:**<br>• 00 - Susclk gating and laneclkreq disabled<br>• 01 - Reserved<br>• 10 - Reserved<br>• 11 - Susclk gating enabled and laneclkreq enabled |
| 4:0 | 0x0<br>RO | **Reserved** |

## 68.2.2 TX_DWORD5 (tx_dword5)—Offset 94h

TX_DWORD5 for HSIO Lane 0 is shown here. This register description applies to the TX_DWORD5 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 94h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 94h

**Default:** 28292A00h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RO | **Reserved** |
| 21:16 | 0x29<br>RW | **ow2tapgen2deemph3p5_5_0:** Tx De-emphasis -3.5dB Setting for Rate1.<br>When Tx is operating in 2-Tap mode, these bits define the 3.5dB de-emphasis setting, i.e. factor (X/64) of slices to use in main-R1.<br>These bits are valid only when rate=2'b01 (typically gen2). |
| 15:14 | 0x0<br>RO | **Reserved** |
| 13:8 | 0x2a<br>RW | **ow2tapgen1deemph3p5_5_0:** Tx De-emphasis -3.5dB Setting for Rate0.<br>When Tx is operating in 2-Tap mode, these bits define the 3.5dB de-emphasis setting, i.e. factor (X/64) of slices to use in main-R1.<br>These bits are valid only when rate=2'b00 (typically gen1). |
| 7:0 | 0x0<br>RO | **Reserved** |

### 68.2.3 TX_DWORD6 (tx_dword6)—Offset 98h

TX_DWORD6 for HSIO Lane 0 is shown here. This register description applies to the TX_DWORD6 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 98h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 98h

**Default:** 151F2020h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:22 | 0x0<br>RO | **Reserved** |
| 21:16 | 0x1f<br>RW | **ow2tapgen3deemph6p0_5_0:** Tx De-emphasis -6.0dB Setting for Rate2.<br>When Tx is operating in 2-Tap mode, these bits define the 6.0dB de-emphasis setting, i.e. factor (X/64) of slices to use in main-R1.<br>These bits are valid only when rate=2'b10 (typically gen3). |
| 15:14 | 0x0<br>RO | **Reserved** |
| 13:8 | 0x20<br>RW | **ow2tapgen2deemph6p0_5_0:** Tx De-emphasis -6.0dB Setting for Rate1.<br>When Tx is operating in 2-Tap mode, these bits define the 6.0dB de-emphasis setting, i.e. factor (X/64) of slices to use in main-R1.<br>These bits are valid only when rate=2'b01 (typically gen2). |
| 7:0 | 0x0<br>RO | **Reserved** |

### 68.2.4 TX_DWORD7 (tx_dword7)—Offset 9Ch

TX_DWORD7 for HSIO Lane 0 is shown here. This register description applies to the TX_DWORD7 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 9Ch
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 9Ch

**Default:** 022A0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved** |
| 29:24 | 0x20 | **Omargom011_5_0:** Tx Margin011 factor (X/64) of slices to use for full swing level. |
| 23:22 | 0x0<br>RO | **Reserved** |
| 21:16 | 0x2A | **Omargom010_5_0:** Tx Margin010 factor (X/64) of slices to use for full swing level. |
| 15:14 | 0x0<br>RO | **Reserved** |
| 13:8 | 0x0 | **Omargom001_5_0:** Tx Margin001 factor (X/64) of slices to use for full swing level. |
| 7:6 | 0x0<br>RO | **Reserved** |
| 5:0 | 0x0 | **Omargom000_5_0:** Tx Margin000 factor (x/64) of slices to use for full swing level. |

## 68.2.5 TX_DWORD19 (tx_dword19)—Offset CCh

TX_DWORD19 for HSIO Lane 0 is shown here. This register description applies to the TX_DWORD19 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + CCh
**Host Memory Space:** SBREG_BAR + 0xBA0000 + CCh

**Default:** 201F0000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:1 | 0x0<br>RO | **Reserved** |
| 0 | 0x0<br>RW | **o_calccont:** initiate calculation of swing-control while '1' the calculation is done consecutively. |

## 68.2.6 RX_DWORD20 (rx_dword20)—Offset 150h

RX_DWORD20 for HSIO Lane 0 is shown here. This register description applies to the RX_DWORD20 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 150h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 150h

**Default:** 003F3F3Fh

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:30 | 0x0<br>RO | **Reserved** |
| 29:24 | 0x0<br>RW | **icfgctledatatap_fullrate_5_0:**<br>FULL rate CTLE Tap Co-efficients override in non-adaptive system. |
| 23:0 | 0x0<br>RO | **Reserved** |

## 68.2.7 RX_DWORD21 (rx_dword21)—Offset 154h

RX_DWORD21 for HSIO Lane 0 is shown here. This register description applies to the RX_DWORD21 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 154h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 154h

**Default:** 80C00000h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:14 | 0x0<br>RO | **Reserved** |
| 13:8 | 0x0<br>RW | **icfgctledatatap_quatrate_5_0:**<br>QUARTER rate CTLE Tap Co-efficients override in non-adaptive system. |
| 7:4 | 0x0<br>RO | **Reserved** |
| 5:0 | 0x0<br>RW | **icfgctledatatap_halfrate_5_0:**<br>HALF rate CTLE Tap Co-efficients override in non-adaptive system. |

## 68.2.8 RX_DWORD25 (rx_dword25)—Offset 164h

RX_DWORD25 for HSIO Lane 0 is shown here. This register description applies to the RX_DWORD25 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 164h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 164h

**Default:** 00005400h

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:21 | 0x0<br>RO | **Reserved** |
| 20:16 | 0x0<br>RW | **ctle_adapt_offset_cfg_4_0:**<br>The 2s complement value to be added to the CTLE coefficient after its training. |
| 15:0 | 0x0<br>RO | **Reserved** |

## 68.2.9 RX_DWORD26 (rx_dword26)—Offset 168h

RX_DWORD26 for HSIO Lane 0 is shown here. This register description applies to the RX_DWORD26 for other 19 HSIO Lanes. See Table 68-2, "Summary of Sideband Registers (Lane 1 of 19) — 0xBA" through Table 68-20, "Summary of Sideband Registers (Lane 19 of 19) — 0xBB" for the applicable Sideband Ports and offset values.

**Type:** Sideband Register
(Size: 32 bits)

**Sideband Port and Offset:** 0xBA + 168h
**Host Memory Space:** SBREG_BAR + 0xBA0000 + 168h

**Default:** B800822Ah

| Bit Range | Default & Access | Field Name (ID): Description |
|---|---|---|
| 31:17 | 0x0<br>RO | **Reserved** |
| 16 | 0x0<br>RW | **sata_eq_dis:** chicken bit disable SATA3 doing equation by default.<br>• 1: disable eq<br>• 0: run eq |
| 15:0 | 0x0<br>RO | **Reserved** |

§ §

# 69 Model-Specific Registers (MSR)

## 69.1 Introduction and Index

The host-accessible registers for the Model-Specific Registers (MSR) are described here.

To determine the number of processor cores, L2 Cache units, and Memory Controllers refer to the following:

- Table 1-1, “Features by product SKU - Server and Cloud Storage SKUs 0 through 4”
- Table 1-2, “Features by Product SKU - Network and Enterprise Storage SKUs 5 through 10”
- Table 1-3, “Features by Product SKU - Extended Temperature SKUs 11 through 14”

The legend for the register access codes are described in Chapter 43, “Introduction.”

## Table 69-1. Model Specific Register (MSR) Index (Sheet 1 of 5)

| MSR Address | MSR Name |
|---|---|
| 006h | (006h) IA32_MONITOR_FILTER_SIZE on page 3801 |
| 017h | (017h) IA32_PLATFORM_ID on page 3802 |
| 01Bh | (01Bh) IA32_APIC_BASE on page 3802 |
| 02Eh | (02Eh) MSR_PIC_MSG_CONTROL on page 3803 |
| 033h | (033h) MSR_MEMORY_CONTROL on page 3803 |
| 034h | (034h) MSR_SMI_COUNT on page 3803 |
| 035h | (035h) CORE_THREAD_COUNT on page 3804 |
| 039h | (039h) MSR_THREAD_ID on page 3804 |
| 03Bh | (03Bh) IA32_TSC_ADJUST on page 3804 |
| 04Eh | (04Eh) MSR_PPIN_CTL on page 3805 |
| 04Fh | (04Fh) MSR_PPIN on page 3805 |
| 052h | (052h) SMM_MCA_CONTROL on page 3805 |
| 079h | (079h) IA32_BIOS_UPDT_TRIG on page 3806 |
| 07Ah | (07Ah) MSR_SGX_FEATURE_ACTIVATION on page 3806 |
| 08Bh | (08Bh) IA32_BIOS_SIGN_ID on page 3806 |
| 09Bh | (09Bh) IA32_SMM_MONITOR_CTL on page 3806 |
| 09Eh | (09Eh) IA32_SMBASE on page 3807 |
| 0C1h | (0C1h) IA32_PMC0 on page 3807 |
| 0C2h | (0C2h) IA32_PMC1 on page 3807 |
| 0C3h | (0C3h) IA32_PMC2 on page 3807 |
| 0C4h | (0C4h) IA32_PMC3 on page 3807 |
| 0CEh | (0CEh) MSR_PLATFORM_INFO on page 3808 |
| 0E2h | (0E2h) MSR_PKG_CST_CONFIG_CONTROL on page 3810 |
| 0E4h | (0E4h) MSR_PMG_IO_CAPTURE_BASE on page 3811 |
| 0E7h | (0E7h) IA32_MPERF on page 3811 |
| 0E8h | (0E8h) IA32_APERF on page 3811 |
| 0FEh | (0FEh) IA32_MTRRCAP on page 3812 |
| 120h | (120h) POWER_MISC on page 3812 |
| 121h | (121h) EMULATE_PM_TMR on page 3813 |
| 139h | (139h) INITIAL_BOOT_BLOCK_COMPLETE on page 3813 |
| 13Ch | (13Ch) FEATURE_CONFIG on page 3813 |
| 140h | (140h) FEATURE_ENABLES on page 3814 |
| 15Ch | (15Ch) PERMEM_CONFIG_INFO on page 3814 |
| 15Dh | (15Dh) PERMEM_CTRLR_INDEX on page 3814 |
| 15Eh | (15Eh) PERMEM_CTRLR_ID on page 3814 |
| 174h | (174h) IA32_SYSENTER_CS on page 3815 |
| 175h | (175h) IA32_SYSENTER_ESP on page 3815 |
| 176h | (176h) IA32_SYSENTER_EIP on page 3815 |
| 179h | (179h) IA32_MCG_CAP on page 3816 |

## Table 69-1. Model Specific Register (MSR) Index (Sheet 2 of 5)

| MSR Address | MSR Name |
|---|---|
| 17Ah | (17Ah) IA32_MCG_STATUS on page 3816 |
| 17Dh | (17Dh) MSR_SMM_MCA_CAP on page 3817 |
| 17Fh | (17Fh) MSR_ERROR_CONTROL on page 3817 |
| 186h | (186h) IA32_PERFEVTSEL0 on page 3818 |
| 187h | (187h) IA32_PERFEVTSEL1 on page 3818 |
| 188h | (188h) IA32_PERFEVTSEL2 on page 3818 |
| 189h | (189h) IA32_PERFEVTSEL3 on page 3819 |
| 198h | (198h) IA32_PERF_STATUS on page 3819 |
| 199h | (199h) IA32_PERF_CTL on page 3819 |
| 19Ah | (19Ah) IA32_CLOCK_MODULATION on page 3820 |
| 19Bh | (19Bh) IA32_THERM_INTERRUPT on page 3820 |
| 19Ch | (19Ch) IA32_THERM_STATUS on page 3821 |
| 19Dh | (19Dh) MSR_THERM2_CTL on page 3822 |
| 1A0h | (1A0h) IA32_MISC_ENABLE on page 3823 |
| 1A1h | (1A1h) PACKAGE_THERM_MARGINBD on page 3824 |
| 1A2h | (1A2h) MSR_TEMPERATURE_TARGET on page 3824 |
| 1A4h | (1A4h) MSR_MISC_FEATURE_CONTROL on page 3824 |
| 1A6h | (1A6h) MSR_OFFCORE_RSP_0 on page 3825 |
| 1A7h | (1A7h) MSR_OFFCORE_RSP_1 on page 3825 |
| 1A8h | (1A8h) THREAD_SW_DEFEATURE on page 3825 |
| 1AAh | (1AAh) MSR_MISC_PWR_MGMT on page 3825 |
| 1ADh | (1ADh) MSR_TURBO_RATIO_LIMIT on page 3826 |
| 1AEh | (1AEh) MSR_TURBO_RATIO_LIMIT1 on page 3826 |
| 1B0h | (1B0h) IA32_ENERGY_PERF_BIAS on page 3826 |
| 1B1h | (1B1h) IA32_PACKAGE_THERM_STATUS on page 3827 |
| 1B2h | (1B2h) IA32_PACKAGE_THERM_INTERRUPT on page 3828 |
| 1C6h | (1C6h) DEBUG_RESOURCE_STATUS on page 3828 |
| 1C8h | (1C8h) MSR_LBR_SELECT on page 3829 |
| 1C9h | (1C9h) MSR_LASTBRANCH_TOS on page 3829 |
| 1D9h | (1D9h) IA32_DEBUGCTL on page 3830 |
| 1E3h | (1E3h) DEBUG_ERR_INJ_CTL on page 3830 |
| 1E4h | (1E4h) DEBUG_ERR_INJ_CTL2 on page 3831 |
| 1E6h | (1E6h) PIC_DEBUG_MODES on page 3832 |
| 1F0h | (1F0h) VLW_CAPABILITY on page 3834 |
| 1FCh | (1FCh) MSR_POWER_CTL on page 3835 |
| 281h | (281h) IA32_MC1_CTL2 (L2 Cache) on page 3836 |
| 282h | (282h) IA32_MC2_CTL2 (FEC) on page 3836 |
| 283h | (283h) IA32_MC3_CTL2 (MEC) on page 3836 |
| 284h | (284h) IA32_MC4_CTL2 (SA) on page 3836 |
| 285h | (285h) IA32_MC5_CTL2 (PMU) on page 3837 |

## Table 69-1. Model Specific Register (MSR) Index (Sheet 3 of 5)

| MSR Address | MSR Name |
|---|---|
| 286h | (286h) IA32_MC6_CTL2 (SA) on page 3837 |
| 287h | (287h) IA32_MC7_CTL2 (MEM 0) on page 3837 |
| 288h | (288h) IA32_MC8_CTL2 (MEM 1) on page 3838 |
| 2E0h | (2E0h) NO_EVICT_MODE on page 3838 |
| 2FFh | (2FFh) IA32_MTRR_DEF_TYPE on page 3838 |
| 300h | (300h) MSR_SGXOWNER0 on page 3839 |
| 301h | (301h) MSR_SGXOWNER1 on page 3839 |
| 309h | (309h) IA32_FIXED_CTR0 on page 3839 |
| 30Ah | (30Ah) IA32_FIXED_CTR1 on page 3839 |
| 30Bh | (30Bh) IA32_FIXED_CTR2 on page 3839 |
| 345h | (345h) IA32_PERF_CAPABILITIES on page 3840 |
| 38Dh | (38Dh) IA32_FIXED_CTR_CTRL on page 3840 |
| 38Eh | (38Eh) IA32_PERF_GLOBAL_STATUS on page 3841 |
| 38Fh | (38Fh) IA32_PERF_GLOBAL_CTRL on page 3842 |
| 390h | (390h) IA32_PERF_GLOBAL_STATUS_RESET on page 3843 |
| 391h | (391h) IA32_PERF_GLOBAL_STATUS_SET on page 3844 |
| 392h | (392h) IA32_PERF_GLOBAL_INUSE on page 3844 |
| 3F1h | (3F1h) IA32_PEBS_ENABLE on page 3845 |
| 3F8h | (3F8h) MSR_PKG_C3_RESIDENCY on page 3845 |
| 3F9h | (3F9h) MSR_PKG_C6_RESIDENCY on page 3845 |
| 3FCh | (3FCh) MSR_CORE_C3_RESIDENCY on page 3845 |
| 3FDh | (3FDh) MSR_CORE_C6_RESIDENCY on page 3846 |
| 3FEh | (3FEh) MSR_CORE_C7_RESIDENCY on page 3846 |
| 400h | (400h) IA32_MC0_CTL (BIU) on page 3846 |
| 401h | (401h) IA32_MC0_STATUS on page 3847 |
| 402h | (402h) IA32_MC0_ADDR (BIU) on page 3848 |
| 404h | (404h) IA32_MC1_CTL (L2 Cache) on page 3848 |
| 405h | (405h) IA32_MC1_STATUS (L2 Cache) on page 3849 |
| 406h | (406h) IA32_MC1_ADDR (L2 Cache) on page 3851 |
| 408h | (408h) IA32_MC2_CTL (FEC) on page 3851 |
| 409h | (409h) IA32_MC2_STATUS (FEC) on page 3852 |
| 40Ah | (40Ah) IA32_MC2_ADDR (FEC) on page 3854 |
| 40Ch | (40Ch) IA32_MC3_CTL (MEC) on page 3854 |
| 40Dh | (40Dh) IA32_MC3_STATUS (MEC) on page 3855 |
| 40Eh | (40Eh) IA32_MC3_ADDR (MEC) on page 3855 |
| 410h | (410h) IA32_MC4_CTL (SA) on page 3856 |
| 411h | (411h) IA32_MC4_STATUS (SA) on page 3858 |
| 412h | (412h) IA32_MC4_ADDR (SA) on page 3860 |
| 413h | (413h) IA32_MC4_MISC (SA) on page 3860 |
| 414h | (414h) IA32_MC5_CTL (PMU) on page 3861 |

## Table 69-1. Model Specific Register (MSR) Index (Sheet 4 of 5)

| MSR Address | MSR Name |
|---|---|
| 415h | (415h) IA32_MC5_STATUS (PMU) on page 3863 |
| 416h | (416h) IA32_MC5_ADDR (PMU) on page 3864 |
| 417h | (417h) IA32_MC5_MISC (PMU) on page 3864 |
| 418h | (418h) IA32_MC6_CTL (SA) on page 3865 |
| 419h | (419h) IA32_MC6_STATUS (SA) on page 3867 |
| 41Ah | (41Ah) IA32_MC6_ADDR (SA) on page 3867 |
| 41Ch | (41Ch) IA32_MC7_CTL (MEM 0) on page 3868 |
| 41Dh | (41Dh) IA32_MC7_STATUS (MEM 0) on page 3870 |
| 41Fh | (41Fh) IA32_MC7_MISC (MEM 0) on page 3871 |
| 420h | (420h) IA32_MC8_CTL (MEM 1) on page 3872 |
| 421h | (421h) IA32_MC8_STATUS (MEM 1) on page 3874 |
| 423h | (423h) IA32_MC8_MISC (MEM 1) on page 3874 |
| 4C1h | (4C1h) IA32_A_PMC0 on page 3874 |
| 4C2h | (4C2h) IA32_A_PMC1 on page 3875 |
| 4C3h | (4C3h) IA32_A_PMC2 on page 3875 |
| 4C4h | (4C4h) IA32_A_PMC3 on page 3875 |
| 4E0h | (4E0h) MSR_SMM_FEATURE_CONTROL on page 3875 |
| 4E2h | (4E2h) MSR_SMM_DELAYED on page 3876 |
| 4E3h | (4E3h) MSR_SMM_BLOCKED on page 3876 |
| 500h | (500h) IA32_SGX_SVN_STATUS on page 3876 |
| 503h | (503h) MSR_SGX_DEBUG_MODE on page 3877 |
| 560h | (560h) IA32_RTIT_OUTPUT_BASE on page 3877 |
| 561h | (561h) IA32_RTIT_OUTPUT_MASK_PTRS on page 3877 |
| 570h | (570h) IA32_RTIT_CTL on page 3878 |
| 571h | (571h) ARR_CR_RTIT_STATUSIA32_RTIT_STATUS on page 3878 |
| 572h | (572h) IA32_RTIT_CR3_MATCH on page 3878 |
| 580h | (580h) IA32_RTIT_ADDR0_A on page 3878 |
| 581h | (581h) IA32_RTIT_ADDR0_B on page 3879 |
| 582h | (582h) IA32_RTIT_ADDR1_A on page 3879 |
| 583h | (583h) IA32_RTIT_ADDR1_B on page 3879 |
| 600h | (600h) IA32_DS_AREA on page 3879 |
| 601h | (601h) MSR_VR_CURRENT_CONFIG on page 3880 |
| 603h | (603h) MSR_VR_MISC_CONFIG on page 3880 |
| 606h | (606h) MSR_RAPL_POWER_UNIT on page 3881 |
| 610h | (610h) MSR_PKG_POWER_LIMIT on page 3882 |
| 614h | (614h) MSR_PKG_POWER_INFO on page 3884 |
| 615h | (615h) PPL3_CONTROL on page 3884 |
| 618h | (618h) MSR_DRAM_POWER_LIMIT on page 3885 |
| 61Ch | (61Ch) MSR_DRAM_POWER_INFO on page 3886 |
| 633h | (633h) MSR_PKGC_IRTL3 on page 3887 |

## Table 69-1. Model Specific Register (MSR) Index (Sheet 5 of 5)

| MSR Address | MSR Name |
|---|---|
| 634h | (634h) MSR_PKGC_IRTL4 on page 3887 |
| 635h | (635h) MSR_PKGC_IRTL5 on page 3888 |
| 639h | (639h) MSR_PP0_ENERGY_STATUS on page 3888 |
| 641h | (641h) MSR_PP1_ENERGY_STATUS on page 3889 |
| 64Ch | (64Ch) MSR_TURBO_ACTIVATION_RATIO on page 3889 |
| 64Fh | (64Fh) MSR_CORE_PERF_LIMIT_REASONS on page 3890 |
| 80Fh | (80Fh) IA32_X2APIC_SIVR on page 3891 |
| 828h | (828h) IA32_X2APIC_ESR on page 3891 |
| 82Fh | (82Fh) IA32_X2APIC_LVT_CMCI on page 3892 |
| 830h | (830h) IA32_X2APIC_ICR on page 3893 |
| 832h | (832h) IA32_X2APIC_LVT_TIMER on page 3894 |
| 833h | (833h) IA32_X2APIC_LVT_THERMAL on page 3895 |
| 834h | (834h) IA32_X2APIC_LVT_PMI on page 3895 |
| 835h | (835h) IA32_X2APIC_LVT_LINT0 on page 3896 |
| 836h | (836h) IA32_X2APIC_LVT_LINT1 on page 3897 |
| 837h | (837h) IA32_X2APIC_LVT_ERROR on page 3897 |
| 838h | (838h) IA32_X2APIC_INIT_COUNT on page 3898 |
| 839h | (839h) IA32_X2APIC_CUR_COUNT on page 3898 |
| 83Eh | (83Eh) IA32_X2APIC_DIV_CONF on page 3898 |
| 83Fh | (83Fh) IA32_X2APIC_SELF_IPI on page 3898 |
| C80h | (C80h) IA32_DEBUG_FEATURE on page 3899 |
| C8Fh | (C8Fh) IA32_PQR_ASSOC on page 3899 |
| D10h | (D10h) IA32_L2_QOS_MASK_0 on page 3899 |
| D11h | (D11h) IA32_L2_QOS_MASK_1 on page 3900 |
| D12h | (D12h) IA32_L2_QOS_MASK_2 on page 3900 |
| D13h | (D13h) IA32_L2_QOS_MASK_3 on page 3900 |
| D90h | (D90h) IA32_BNDCFGS on page 3901 |
| DA0h | (DA0h) IA32_XSS on page 3901 |
| C0000080h | (C0000080h) IA32_EFER on page 3901 |
| C0000081h | (C0000081h) IA32_STAR on page 3902 |
| C0000082h | (C0000082h) IA32_LSTAR on page 3902 |
| C0000083h | (C0000083h) IA32_CSTAR on page 3902 |
| C0000084h | (C0000084h) IA32_FMASK on page 3902 |
| C0000100h | (C0000100h) IA32_FS_BASE on page 3903 |
| C0000101h | (C0000101h) IA32_GS_BASE on page 3903 |
| C0000102h | (C0000102h) IA32_KERNEL_GS_BASE on page 3903 |
| C0000103h | (C0000103h) IA32_TSC_AUX on page 3903 |

## 69.2 Model Specific Registers (MSRs)

### 69.2.1 (006h) IA32_MONITOR_FILTER_SIZE

OS or BIOS can program to the maximum line size in the system. CPUID will return the maximum line size of the processor and the system for the purposes of Monitor and MWait. Software can use this information to pad data appropriately to avoid cross-cluster hot-spots.

| MSR Address: 006h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Thread | 0h | RW | Reserved. |
| 15:0 | Thread | 0h | RW | **FILTER_MAXIMUM_LINE_SIZE —** OS or BIOS can program to the maximum line size in the system. CPUID will return the maximum line size of the processor and the system for the purposes of Monitor and MWait. Software can use this information to pad data appropriately to avoid cross-cluster hot-spots. |

### 69.2.2 (017h) IA32_PLATFORM_ID

Used for selecting which patch to use. Indicates the platform that the processor is intended for. This MSR is used for microcode update loading purpose only. For all other device identification purposes it is recommended that System BIOS use PCI based Device Identification register.

| MSR Address: 17h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:53 | Package | 0h | RSV | Reserved. |
| 52:50 | Package | 0h | RW | **PLATFORM_ID —** This field is loaded from SoC fuses. |
| 49:0 | Package | 0h | RSV | Reserved. |

### 69.2.3 (01Bh) IA32_APIC_BASE

Register used to implement the APICBASE MSR. Contains the base address for all the APIC registers along with certain APIC control and state bits. The xAPIC Base Register holds the APIC base address, permitting the relocation of the APIC memory map.

| MSR Address: 01Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:40 | Thread | 0h | Rsvd0 | Reserved. |
| 39:12 | Thread | 0h | RW | **APIC_BASE —** Bits 39:12 of the physical APIC register base address. WRMSR will GP fault on writes of 1 to 35:32 for PA32 parts. |
| 11 | Thread | 0h | RW | **APEN —** APIC enable bit. If bit is 0, then the SoC APIC is disabled. If the bit is 1, then APIC is enabled. |
| 10 | Thread | 0h | RW | **EXTENDED_MODE_ENABLE —** Enable/Disable for APIC Extensions. |
| 9 | Thread | 0h | Rsvd0 | Reserved. |
| 8 | Thread | 0h | RW | **BSP —** Bit used to indicate that the processor is the Boot Strap Processor (BSP). BIOS must clear this bit if the processor is not going to be the System BSP (SBSP). |
| 7:0 | Thread | 0h | Rsvd0 | Reserved. |

### 69.2.4 (02Eh) MSR_PIC_MSG_CONTROL

MSR used to enable/disable TPR messages. See NHM-B bug 2425138. Controls APIC TPR Update messages from the processor's Local APIC. By default at reset, TPR Update messages are disabled so they are not queued up while not being serviced. However, it is the BIOS' responsibility to enable TPR Update messages before any Local APIC mode transitions occur.

| MSR Address: 02Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:11 | - | - | - | Reserved. |
| 10 | Module | 0h | RO | **TPR_MSG_OFF —** Prevents all TPR / IntPrioUpd messages from being generated, including those generated from APIC ID/LogID/DFR writes and Enable changes. WRMSR only allows clearing of this bit. Any attempt to set it (if already clear) will be ignored. |
| 9:0 | - | - | - | Reserved. |

### 69.2.5 (033h) MSR_MEMORY_CONTROL

DCU_TEST_CTL MSR. Ucode mirrors the appropriate split lock disable bit and this entry is used only to generate the proper address defines. This register is used to disable split locks, which are locked instructions that split a cache line.

| MSR Address: 033h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Thread | 0h | RW | **SPLIT_LOCK_DISABLE —** If set to 1, disables split locks. If clear to 0, enables split locks. Setting this bit will cause split locks to be non-atomic, despite the lock attribute of the instruction. |
| 30:0 | - | - | - | Reserved. |

### 69.2.6 (034h) MSR_SMI_COUNT

SLM PerfMon:Number of times SMISystem Management Mode (SMM) was entered.

| MSR Address: 034h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | 0h | RW | Reserved. |
| 31:0 | Thread | 0h | RO | **SMM_ENTRY_COUNT —** Holds a running count of SMI events since the last reset. |

### 69.2.7 (035h) CORE_THREAD_COUNT

This MSR is shared in the processor package and used to determine the current count of enabled Cores and Threads. This register reflects the actual number of enabled Cores and Threads based on the factory configurations and BIOS modifications. The Core Count and ThreadCount fields reflect the enabled cores and threads based on the factory-configured core.

| MSR Address: 035h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:16 | Package | - | RO | **CORE_COUNT —** The Core Count reflects the enabled cores based on the factory-configured core count and the value of the RESOLVED_CORES_MASK register at reset time. |
| 15:0 | Package | - | RO | **THREAD_COUNT —** The Thread Count reflects the enabled threads based on the factory-configured thread count and the value of the RESOLVED_CORES_MASK register at reset time. |

### 69.2.8 (039h) MSR_THREAD_ID

On P4, external software is allowed to set bits 31:5 to an arbitrary value to support clustered systems.

| MSR Address: 039h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:8 | Thread | 0h | Rsvd0 | **RESERVED —** reserved |
| 7:0 | Thread | 0h | RW | **APIC_ID —** From the local APIC ID. Read only unless PIC debug mode is enabled. |

### 69.2.9 (03Bh) IA32_TSC_ADJUST

UTSC_THREAD_OFFSET is Used to read the thread specific offset to Time Stamp counter base value in RDTSC computation. This helps in synchronizing between the cores for TSC calibration. This feature is available only when CPUID.(EAX=7):EBX[1] is 1.

| MSR Address: 03Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **THREAD_OFFSET —** A read from this MSR would return the thread's offset to TSC base value, that is used in computing RDTSC return value. A write to this MSR would write only the thread's offset to the TSC value and not update the actual counter value. |

## 69.2.10 (04Eh) MSR_PPIN_CTL

Protected Processor Inventory Number (PPIN) Enable Control The reset value is 0. It will not survive warm reset.

| MSR Address: 04Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Package | 0h | RW | **ENABLE —** Indication to SW about presence of PPIN<br>0 - PPIN not present. Read of PPIN MSR results in a #GP.<br>1 - PPIN present. Read of PPIN MSR by SW is allowed. |
| 0 | Package | 0h | RW | **LOCK —** Bit indicates if PPIN_CTL is locked<br>0 - Not locked (SW free to write to PPIN_CTL).<br>1 - Locked (SW can no longer write to PPIN_CTL). |

## 69.2.11 (04Fh) MSR_PPIN

MSR for PPIN feature. Access to MSR_PPIN is permitted only if MSR_PPIN_CTL[1:0]='10b'

| MSR Address: 4Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RO | **VALUE —** The 64 bit inventory number. |

## 69.2.12 (052h) SMM_MCA_CONTROL

Allows System Management mode (SMM) firmware to intercept Corrected and UCNA (Uncorrected, No Action) errors.

| MSR Address: 052h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:10 | - | - | - | Reserved. |
| 9 | Thread | 0h | RW | **PEND_SMI_ON_MCA —** Pend SMI on MCA |
| 8:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | rsvd0 | **UCNA_RD_STATUS_IN_SMM_ONLY —** The SoC design has no UCNA errors. |
| 0 | Thread | 0h | RW | **CERR_RD_STATUS_IN_SMM_ONLY —** Only read valid correctable errors status when in SMM. Otherwise read 0 if V=1 and UC=0. |

## 69.2.13 (079h) IA32_BIOS_UPDT_TRIG

Used to trigger the loading of the Microcode Update to the processor. Refer to the 'Microcode Update' section of the BIOS Writer's Guide for further information on the Microcode Update loading. Attempts to read this MSR will result in a GP fault. The IA32_BIOS_UPDT_TRIG register is not shared across cores in the same physical processor. BIOS must load the microcode update for each thread in a processor package.

| MSR Address: 079h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | 0h | WO | **DATA —** Any data write will cause the microcode update loader to get triggered. |

## 69.2.14 (07Ah) MSR_SGX_FEATURE_ACTIVATION

| MSR Address: 07Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:1 | - | - | - | Reserved. |
| 0 | Core | 0h | RW | **SE —** Secure Enclaves Feature Activation |

## 69.2.15 (08Bh) IA32_BIOS_SIGN_ID

Microcode Update Signature.

| MSR Address: 08Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | 0h | RW | **PATCH_SIGN_ID —** Microcode Update Signature. |
| 31:0 | Thread | 0h | Rsvd0 | Reserved. |

## 69.2.16 (09Bh) IA32_SMM_MONITOR_CTL

This MSR is writeable only in System Management Mode (SMM), else the SoC generates #GP.

| MSR Address: 09Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | 0h | RW | Reserved. |
| 31:12 | Thread | 0h | RW | **MSEG_BASE_PAGE_FRAME_ADR —** VMX MSEG base page frame address |
| 11:3 | Thread | 0h | RW | Reserved. |
| 2 | Thread | 0h | RW | **VMXOFFCTRL —** VMX Off Control. When this bit is set, VMXOFF will unmask SMI events. |
| 1 | Thread | 0h | RW | **CSTATE_SMI_OPTIN —** C-state SMI VMX opt in. Can only be set by WRMSR if IA32_FEATURE_CONTROL_MSR.C_STATE_SMI and LOCK |
| 0 | Thread | 0h | RW | **VALID —** When set, this bit indicates that the MSR contents are valid. |

### 69.2.17 (09Eh) IA32_SMBASE

Read only MSR. WRMSR will always #GP. RDMSR will GP if not in SMM mode.

| MSR Address: 09Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | - | - | - | **Reserved (RSVD) —** Reserved. |
| 31:0 | Thread | 0h | RO | **DATA —** Register which contains the current SMM base value for the processor. The RESET event handler should set it to 0x00030000, which is the default value. When RSM executes, it reads the SMM base slot value in SMM memory and loads it into this register. |

### 69.2.18 (0C1h) IA32_PMC0

Performance Counter Register 0. This register is available only if CPUID (EAX=0Ah):EAX[15:8] > 0.

| MSR Address: 0C1h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | - | - | **DATA** — Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

### 69.2.19 (0C2h) IA32_PMC1

Performance Counter Register 1. This register is available only if CPUID (EAX=0Ah):EAX[15:8] > 1.

| MSR Address: 0C2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | - | - | **DATA** — Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

### 69.2.20 (0C3h) IA32_PMC2

Performance Counter Register 2. This register is available only if CPUID (EAX=0Ah):EAX[15:8] > 2.

| MSR Address: 0C3h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | - | - | **DATA** — Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

### 69.2.21 (0C4h) IA32_PMC3

Performance Counter Register 3. This register is available only if CPUID (EAX=0Ah):EAX[15:8] > 3.

| MSR Address: 0C4h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | - | - | **DATA** — Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

## 69.2.22 (0CEh) MSR_PLATFORM_INFO

This contains information about platforms frequency capabilities that the CPU reports.

| MSR Address: CEh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:56 | Package | 0h | RO | Reserved. |
| 55:48 | Package | 1h | RW | **MIN_OPERATING_RATIO —** Min supported ratio in units of 100MHz Confirm with Jeff |
| 47:40 | Package | 0h | RW | **MAX_EFFICIENCY_RATIO —** Maximum Efficiency Ratio. This is given in units of 100 MHz. |
| 39:38 | Package | 0h | RO | Reserved. |
| 37 | Package | 0h | RW | **TIMED_MWAIT_ENABLE —** Timed MWAIT enable |
| 36 | Package | 0h | RO | Reserved. |
| 35 | Package | 0h | RW | **PFAT_ENABLE —** Platform Firmware Armoring Technology (PFAT).<br>0 indicates the PFAT feature is not available.<br>1indicates the PFAT feature is available. |
| 34:33 | Package | 0h | RW | **CONFIG_TDP_LEVELS —** Configurable TDP. Indicates the number of configurable TDP levels supported.<br>00 - Config TDP not supported<br>01 - One additional TDP level supported<br>10 - Two additional TDP levels supported |
| 32 | Package | 0h | RW | **LPM_SUPPORT** — Low Power Mode Support. When set to 1, indicates that BIOS may program IA32_PERF_CTL to levels below the max efficiency ratio down to Minimum Ratio bits [55:48].<br>0 - Low Power mode not supported<br>1 - Low Power mode supported Determined by fuse. |
| 31 | Package | 1h | RW | **CPUID_FAULTING_EN —** CPUID Faulting Capability. When set to 1 indicates that the processor supports raising a #GP if CPUID is executed when not in SMM and the CPL > 0. When this bit is set, it indicates that MSR 140h FEATURE_ENABLES bit 0 can be written by a VMM. |
| 30 | Package | 1h | RW | **PRG_TJ_OFFSET_EN —** Programmable TCC Activation Offset. When set to 1, indicates that the TCC Activation Offset field in MSR_TEMPERATURE_TARGET is valid and programmable. When set to 0, indicates it's not programmable. When this bit is 0, an attempt to write to MSR_TEMPERATURE_TARGET bits [27:24] will result in a GP fault. |
| 29 | Package | 1h | RW | **PRG_TDP_LIMIT_EN —** Programmable TDP Limits for Turbo Mode. When set to 1, indicates that TDP Limits for Turbo mode are programmable, and when set to 0, indicates TDP Limits for Turbo mode are not programmable. When this bit is 0, an attempt to write to PP0_POWER_LIMIT, PP1_POWER_LIMIT and PACKAGE_POWER_LIMIT MSR will result in a GP fault. |
| 28 | Package | 1h | RW | **PRG_TURBO_RATIO_EN —** Programmable Ratio Limits for Turbo Mode. When set to 1, indicates Programmable Ratio Limits for Turbo Mode are enabled, and when set to 0, indicates Programmable Ratio Limits for Turbo Mode are disabled. |
| 27 | Package | 0h | RW | **SAMPLE_PART —** Sample. A value of 1 indicates the processor is a preproduction sample and a property of Intel, a value of 0 indicates the part is intended for production. |
| 26 | Package | 0h | RW | **DCU_16K_MODE_AVAIL —** DCU Mode Select Support. When set to 1. indicates that DCU Mode Selection is supported, and when set to 0, indicates that DCU Mode Selection is not available and the processor will remain in the default DCU mode. |
| 25 | Package | 0h | RO | Reserved. |
| 24 | Package | 0h | RW | **OCVOLT_OVRD_EN —** When set to 1, indicates that this part supports Voltage override overclocking. When this bit is 0 the processor does not support overclocking Voltage overrides. |
| 23 | Package | 0h | RW | **PPIN_CAP —** When set to 1, indicates that Protected Processor Inventory Number (PPIN) feature is supported. |
| 22:17 | Package | 0h | RO | Reserved. |

| MSR Address: CEh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 16 | Package | 1h | RO | **SMM_SAVE_CAP —** SMM Save Capability. When set to 1 indicates this feature exists. |
| 15:8 | Package | 0h | RW | **MAX_NON_TURBO_LIM_RATIO —** Maximum Non Turbo Ratio. Contains the max non-turbo ratio. This is the ratio of the frequency that invariant TSC runs at. Frequency = ratio * 100 MHz.<br>***Note:*** The Maximum Non-Turbo Ratio is adjusted to the flexible limit ratio (as specified in the FLEX_RATIO MSR 194h FLEX_RATIO field bits [15:8]) if flexible limit is enabled (by setting the FLEX_RATIO MSR 194h FLEX_EN field bit[16]=1.<br>In case of Configurable TDP feature, the maximum of the available TDP levels is reported in this field. |
| 7:0 | Package | 0h | RO | Reserved. |

## 69.2.23 (0E2h) MSR_PKG_CST_CONFIG_CONTROL

Power Management - C-states configuration. Note the presence of the LOCK bit (15). When set, locks bits 15-0 of this register for further writes, until the next reset occurs.

| MSR Address: 0E2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Core | - | - | Reserved. |
| 31 | Core | 0h | RW | **TIMED_MWAIT_ENABLE —** When set, enables Timed MWAIT feature. When this bit is cleared, all MWAIT timer configurations issued as a part of MWAIT will raise a #GP exception. |
| 30 | Core | 0h | RW | **ENPKGCUNDEMOTION —** Enable Package Cstate Un-Demotion. Its the enable to consider cases where demotion was the incorrect decision in determining Cstate depth. |
| 29 | Core | 0h | RW | **ENPKGCAUTODEMOTION —** Enable Package Cstate Auto-Demotion. It enables use of history of past pkg Cstate depth and residence, as a factor in determining Cstate depth. |
| 28 | Core | 0h | RW | **ENC1UNDEMOTION —** Enable Un-Demotion from Demoted C1. |
| 27 | Core | 0h | RW | **ENC3UNDEMOTION —** Enable Un-Demotion from Demoted C3. |
| 26 | Core | 0h | RW | **C1_STATE_AUTO_DEMOTION_ENABLE —** When set, SoC will conditionally demote C3/C6/C7 requests to C1 based on uncore auto-demote information. |
| 25 | Core | 0h | RW | **C3_STATE_AUTO_DEMOTION_ENABLE —** When set, SoC will conditionally demote C6/C7 requests to C3 based on uncore auto-demote information. |
| 24:16 | Core | - | - | Reserved. |
| 15 | Core | 0h | RW-O | **CST_CFG_LOCK —** When set, locks bits [15:0] of this register for further writes, until the next reset occurs. |
| 14:11 | Core | - | - | Reserved. |
| 10 | Core | 0h | RW-L | **IO_MWAIT_REDIRECTION —** When set, will map I/O_read instructions sent to I/O registers at MSR_PMG_IO_CAPTURE_BASE[15:0] to MWAIT instructions. |
| 9:8 | Core | - | - | Reserved. |
| 7:4 | Core | 0h | RW | **MAX_CORE_C_STATE —** The possible values are:<br>0000 - Reserved<br>0001 - C1<br>0010 - C3<br>0011 - C6 |
| 3:0 | Core | 0h | RW-L | **PACK_LIM —** Limits the resolved package Cstate to be less than or equal to the encoded state:<br>• 0x0 = No Pkg States [Self-Refresh]<br>• 0x1 = No S0Ix [L2 Voltage Reduction]<br>• 0x2 = No Limit [Default]<br><br>This field will also indirectly impact the possible module Cstate. A limit of C1 means that the module cannot leave C0, as C1 is a core-only Cstate. C3 and C6, will restrict the possible module Cstate to that respective value. If the state is unlimited or C7 and deeper, then the module can enter C7. The states C8, C9, and C10 and synonyms for S0i1, S0i2, and S0i3, respectively. |

### 69.2.24 (0E4h) MSR_PMG_IO_CAPTURE_BASE

Power Management I/O base address for I/O trapping to C-states.

See http://biosbits.org.

| MSR Address: 0E4h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:19 | Core | - | - | Reserved. |
| 18:16 | Core | 0h | RW | **CST_RANGE —** The I/O-port block size in which I/O-redirection will be executed (0-7). Should be programmed based on the number of LVLx registers existing in the SoC. |
| 15:0 | Core | 0h | RW | **LVL2B —** The SoC compares I/O-read zone to this base address to determine if an MWAIT(C2/3/4) needs to be issued instead of the I/O-read. Should be programmed to the SoC Plevel_2 I/O address. |

### 69.2.25 (0E7h) IA32_MPERF

Maximum Performance Frequency Clock Count register. IA32_MPERF and IA32_APERF MSRs do not have any meaning by themselves separately. Only IA32_APERF/ IA32_MPERF ratio is architectural.

| MSR Address: 0E7h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **C0_MCNT —** C0 maximum-frequency clock count: Increments at maximum clock frequency (as allowed by the Resolved Ratio) when core is in C0. Cleared by a write to this field. |

### 69.2.26 (0E8h) IA32_APERF

Actual Performance Frequency Clock Count register. IA32_MPERF and IA32_APERF MSRs do not have any meaning by themselves separately. Only IA32_APERF/ IA32_MPERF ratio is architectural.

| MSR Address: 0E8h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **C0_ACNT —** C0 actual-frequency clock count: Accumulates core clock counts at the coordinated clock frequency when core is in C0. Cleared by a write to this field. |

## 69.2.27 (0FEh) IA32_MTRRCAP

Read-only register containing MTRR feature identification for the processor. Should be used in conjunction with the CPUID[EDX] bit 12 to determine MTRR capabilities. This MSR is read only. The SoC generates #GP on any write.

| MSR Address: 0FEh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:13 | Core | - | - | Reserved. |
| 12 | Core | 0h | RO | **PRMRR —** PRMRRs are supported: Reports 1 if PRMRR is fuse enabled. |
| 11 | Core | 0h | RO | **SMRR —** SMRR: 1 indicates supported. |
| 10 | Core | 0h | RO | **WRITE —** Write-Combining memory type supported if set. |
| 9 | Core | - | - | Reserved. |
| 8 | Core | 0h | RO | **FIX —** Fixed range registers supported if set. |
| 7:0 | Core | 0h | RO | **VCNT —** Number of variable range registers. |

## 69.2.28 (120h) POWER_MISC

Miscellaneous power controls used for eHALT enables.

| MSR Address: 120h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:9 | - | - | - | Reserved. |
| 8 | Thread | 0h | RW | **WBINVAL_DL1 —** Used by BIOS to flush DL1->L2. |
| 7 | - | - | - | Reserved. |
| 6 | Thread | 0h | RW | **ENABLE_IA_UNTRUSTED_MODE —** Post boot, BIOS will set this boot to clear the default IA_Boot mode and enter IA Untrusted Mode. |
| 5:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | RW | **ENABLE_INDP_AUTOCM —** Enable Independent Pstate Auto Clock Modulation: Writing 1 to this bit will allow automatic clock modulation based on it's core specific requested ratio if requested ratio is lower than the resolved module ratio. This setting does not apply if requested ratio is lower than minimum ratio supported by core PLL. |
| 2 | Thread | 0h | RW | **ENABLE_ULFM_AUTOCM —** Enable UltraLowFreqMode Auto Clock Modulation: Writing 1 to this bit will allow automatic clock modulation based on it's core specific requested ratio if the resolved module ratio equals the minimum ratio supported by core PLL. |
| 1:0 | - | - | - | Reserved. |

## 69.2.29 (121h) EMULATE_PM_TMR

Emulate PM_TMR.

| MSR Address: 121h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | 0h | RW | **MULTIPLIER —** Constant programmed by BIOS to translate CTC to target frequency (14.31818/4). |
| 31:17 | - | - | - | Reserved. |
| 16 | Thread | 0h | RW | **PM_TMR_BLK_VALID —** Indicates that the MSR has been programmed by BIOS, and thus enables internal emulation of the PM_TMR. |
| 15:0 | Thread | 0h | RW | **PM_TMR_BLK_BASE —** Base I/O port address of the PM_TMR. |

## 69.2.30 (139h) INITIAL_BOOT_BLOCK_COMPLETE

Write-only MSR. WRMSR causes #GP if SECURE_BOOT_CAPABILITY_ENABLE == 0.

| MSR Address: 139h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:1 | - | - | - | Reserved. |
| 0 | Thread | 0h | WO | **BIOS_RDY_FOR_MISC_EVENTS —** WMRSR ignores writes of 0. A write of 1 will set CPU_BIOS_RDY_FOR_MISC_INTERRUPTS in SEC. |

## 69.2.31 (13Ch) FEATURE_CONFIG

This MSR is visible only if CPUID.(EAX=01h):ECX[25]=1, which indicates the processor was manufactured with AES-NI enabled. Note that when AES-NI is disabled, a #UD exception will be raised if software attempts to execute one of the AES-NI instructions.

| MSR Address: 13Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Core | 0h | RW | **AESNI_DISABLE —** This bit disables Advanced Encryption Standard feature on this processor core. To disable AES, System BIOS must write 11b to this MSR. Writes of 00b, 01b or 10b to this MSR will result in AES enable. WRMSR will GP if the lock bit is set. |
| 0 | Core | 0h | RW | **AESNI_LOCK —** 1 = Locked. When set, locks this MSR from being written, writes o this bit will result in GP(0) until an S5 reset occurs. BIOS must ensure this lock bit is set before boot to OS. |

## 69.2.32 (140h) FEATURE_ENABLES

Miscellaneous enables for thread-specific features.

| MSR Address: 140h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:1 | - | - | - | Reserved. |
| 0 | Thread | 0h | RW | **CPUID_GP_ON_CPL_GT_0 —** Causes CPUID to #GP if current privilege level (CPL) is greater than 0 and not in SMM.<br>BIOS should not set this bit. This bit should only be configured by a VMM. |

## 69.2.33 (15Ch) PERMEM_CONFIG_INFO

Specifies the maximum number of memory controllers on the platform capable of supporting memory persistence.

| MSR Address: 15Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:8 | - | - | - | Reserved. |
| 7:0 | Thread | 0h | RO | **NUMBER_OF_PCOMMIT_ADDRESSES —** Maximum number of different PCOMMIT addresses. |

## 69.2.34 (15Dh) PERMEM_CTRLR_INDEX

Index into the list of persistent memory addresses which is accessed when the PERMEM_CTRLR_ID MSR is read/written.

| MSR Address: 15Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Thread | 0h | RW | **LOCK —** When set, WRMSR to PERMEM_CTRL_ID will #GP. WRMSR to clear this bit outside SMM will #GP. That means that once locked, it can be unlocked only from SMM. This bit is cleared on reset. |
| 62:2 | - | - | - | Reserved. |
| 1:0 | Thread | 0h | RW | **INDEX —** Can be programmed to any value in the range 0 to N-1 where N is the value reported in 7:0 of PERMEM_CONFIG_INFO. For the SoC, N=4, and so this is a 2-bit field to allow the remaining fields to be checked by the MSR pla. |

## 69.2.35 (15Eh) PERMEM_CTRLR_ID

Specifies the maximum number of memory controllers on the platform capable of supporting memory persistence.

| MSR Address: 15Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:40 | - | - | - | Reserved. |
| 39:6 | Thread | 0h | RW | **IDENTIFIER —** Value written to this field uniquely identifies a persistent memory controller, whose index is in PERMEM_CTRLR_INDEX. Unused indices must be programmed with a reserved value of 0. |
| 5:0 | - | - | - | Reserved. |

## 69.2.36 (174h) IA32_SYSENTER_CS

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 174h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | - | - | - | Reserved. |
| 31:0 | Thread | 0h | RW | **DATA —** To be defined in the Intel® 64 and IA-32 Architectures Software Developer's Manual. |

## 69.2.37 (175h) IA32_SYSENTER_ESP

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 175h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** To be defined in the Intel® 64 and IA-32 Architectures Software Developer's Manual. |

## 69.2.38 (176h) IA32_SYSENTER_EIP

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 176h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** To be defined in the Intel® 64 and IA-32 Architectures Software Developer's Manual. |

## 69.2.39 (179h) IA32_MCG_CAP

MCA Capability Control Register. This MSR is read only. The SoC generates #GP on any write.

| MSR Address: 179h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:24 | - | - | - | Reserved. |
| 23:16 | Thread | 0h | RO | **NUMBER_OF_EXTENDED_MACHINE_CHECK_REGISTERS —** Indicates the number of extended machine check state registers present on this processor. |
| 15:12 | - | - | - | Reserved. |
| 11 | Thread | 0h | RO | **MCG_YELLOW_GREEN —** Indicates that this processor implements "yellow/ green" health reporting. |
| 10 | Thread | 0h | RO | **MCB_CMCI_P —** Correct MC Error counting / signaling is supported. |
| 9 | Thread | 0h | RO | **EXTENDED_MACHINE_CHECK_MSRS_SUPPORTED —** Indicates that this processor implements the extended machine check state registers found starting at MSR address 180H. The SoC does not implement these extended registers. |
| 8 | Thread | 0h | RO | **IMCG_CTL_IS_PRESENT —** Indicates that this processor implements the IA32_MCG_CTL MSR. The SoC does not implement this MSR. |
| 7:0 | Thread | 0h | RO | **NUMBER_OF_HARDWARE_REPORTING_BANKS_AVAILABLE —** Indicates the number of error-reporting banks available on this processor. |

## 69.2.40 (17Ah) IA32_MCG_STATUS

The Machine Check Status register is used to determine the current state of the machine check architecture of the processor after an error has occurred.

| MSR Address: 17Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:3 | - | - | - | Reserved. |
| 2 | Thread | 0h | RW | **MCIP —** Machine Check In Progress. When set, indicates that a machine check has been generated. If a second machine check is detected while this bit is still set, the processor will enter a shutdown state. Software can write this bit, and should write it to 0 after processing a machine check exception. |
| 1 | Thread | 0h | RW | **EIPV —** Error IP valid. When set, indicates that the instruction pointed to by the instruction pointer pushed on the stack when the machine check was generated is directly associated with the error. |
| 0 | Thread | 0h | RW | **RIPV —** Restart IP valid. When set, indicates that the program execution can be restarted reliably at the instruction pointed to by the instruction pointer pushed on the stack when the machine check was generated. When cleared, the program cannot be reliably restarted. |

## 69.2.41 (17Dh) MSR_SMM_MCA_CAP

This register is only accessible while in SMM. Attempting to read or write this register outside of SMM will cause a #GP exception to be raised. This register indicates support for enhanced SMM features. It also reports which Machine-Check banks support enhanced MCA.

| MSR Address: 17Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:60 | - | - | - | Reserved. |
| 59 | Thread | 0h | SMM-RO | **LONG_FLOW_INDICATION —** Set to indicate SMM long flow indicator support. |
| 58 | Thread | 0h | SMM-RO | **SMM_CODE_ACCESS_CHK —** Set to 1 if the SMM code access check feature is supported and the SMM_FEATURE_CONTROL register (MSR 4E0h) is supported. |
| 57 | Thread | 0h | SMM-RO | **SMM_CPU_SVRSTR —** 0 for the SoC. Set to 1 if the SAVE/RESTORE feature is supported and the SMM_FEATURE_CONTROL register (MSR 4E0h) is supported. |
| 56 | Thread | 0h | SMM-RO | **TARGETED_SMI —** 0 for the SoC. Set to 1 if the thread supports targeted SMI and the SMM_ENABLE register (MSR 4E1h) is supported. |
| 55 | Thread | 0h | SMM-RO | **ERROR_SPOOFING —** 0 for the SoC. Set if Error Spoofing and Enhanced MCA are supported. |
| 54 | Thread | 0h | SMM-RO | **SMM_PROT_MODE —** 0 for the SoC. Set if SMM Protected Mode is supported. |
| 53:32 | - | - | - | Reserved. |
| 31:0 | Thread | 0h | SMM-RO | **BANK_SUPPORT —** ENHANCED_MCA only, so 0 for the SoC. One bit per bank. If bit is set, then the corresponding bank supports enhanced MCA. |

## 69.2.42 (17Fh) MSR_ERROR_CONTROL

Error Control Register. This register allows BIOS to control certain error logging features and their behaviors on the processor.

A #GP Fault will occur when doing a read of this MSR and the ERROR_CONTROL MSR is not available.A non-architectural, package-scoped MSR intended for use by BIOS. The bit will be saved and restored in C7+ states. The reset value for the bit is 0. It will not survive warm reset.

| MSR Address: 17Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:5 | - | - | - | Reserved. |
| 4 | Package | 0h | RW | **CMCI_DISABLE —** When set to 1, disables corrected machine check interrupt entirely, cleared upon each reset. BIOS may use this bit to disable CMCI. Typically useful if OEM offers BIOS based PFA (Predicted Failure Analysis) or "Health Monitoring" capability. BIOS will require error polling, e.g., periodic SMI. Must be set during the BIOS initialization.<br>Once it is set:<br>• Any read to IA32_MCG_CAP[10] bit will result in 0. Indicating that processor does not support CMCI.<br>Any read to IA32_MCi_CTL2[30] bit will result in 0. Indicating that CMCI signaling is not available in machine check bank 'i'.Any write to it will result in no change (read-only). |
| 3:0 | - | - | - | Reserved. |

## 69.2.43 (186h) IA32_PERFEVTSEL0

Performance Event Selection Register 0. This register is available only if CPUID. (EAX=0Ah):EAX[15:8] > 0

| MSR Address: 186h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | - | - | - | Reserved. |
| 31:24 | Thread | 0h | RW | **COUNT_MASK —** When CMASK is not zero, the corresponding performance counter increments each cycle if the event count is greater than or equal to the CMASK. |
| 23 | Thread | 0h | RW | **INVERT_FLAG —** Invert the CMASK. |
| 22 | Thread | 0h | RW | **ENABLE_FLAG —** When set, performance counting is enabled in the performance-monitoring counter; when clear, the counter is disabled. |
| 21 | Thread | 0h | RW | **MYTHREAD_ALLTHREAD —**<br>0 = Count events associated with this logical processor<br>1 = Count event on any logical processor in that core, or for uncore events, any logical processor in the module. |
| 20 | Thread | 0h | RW | **INT_APIC_ENABLE_FLAG —** When set, the processor generates an exception through its local APIC on counter overflow for this counter's thread. |
| 19 | Thread | 0h | RW | **PIN_CONTROL_FLAG —** Enables pin control. |
| 18 | Thread | 0h | RW | **EDGE_DETECT_FLAG —** Enables edge detection if set. |
| 17 | Thread | 0h | RW | **OS_MODE_FLAG —** Counts while in privilege level is ring 0. |
| 16 | Thread | 0h | RW | **USER_MODE_FLAG —** Counts while in privilege level is not ring 0. |
| 15:8 | Thread | 0h | RW | **EVENT_MASK —** Further qualifies the event selected in the event select field.of cache states |
| 7:0 | Thread | 0h | RW | **EVENT_ID —** Selects the event to be monitored |

## 69.2.44 (187h) IA32_PERFEVTSEL1

Performance Event Selection Register 0. This register is available only if CPUID. (EAX=0Ah):EAX[15:8] > 1

| MSR Address: 187h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** Refer to the description of IA32_PERFEVTSEL0. |

## 69.2.45 (188h) IA32_PERFEVTSEL2

Performance Event Selection Register 0. This register is available only if CPUID. (EAX=0Ah):EAX[15:8] > 2

| MSR Address: 188h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** Refer to the description of IA32_PERFEVTSEL0. |

## 69.2.46 (189h) IA32_PERFEVTSEL3

Performance Event Selection Register 0. This register is available only if CPUID. (EAX=0Ah):EAX[15:8] > 3

| MSR Address: 189h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** Refer to the description of IA32_PERFEVTSEL0. |

## 69.2.47 (198h) IA32_PERF_STATUS

Performance Status MSR.

| MSR Address: 198h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | - | - | - | Reserved. |
| 47:32 | Module | 0h | RO | **CORE_VOLTAGE —** Voltage in 3.13 format. |
| 31:16 | - | - | - | Reserved. |
| 15:8 | Module | 0h | RO | **PLLRATIO —** Current performance State Value bits 15:8.Ratio of the core pll. |
| 7:0 | - | - | - | Reserved. |

## 69.2.48 (199h) IA32_PERF_CTL

Performance Control MSR. SW makes request for a new Performance state (P-State) by writing this MSR.

| MSR Address: 199h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:33 | - | - | - | Reserved. |
| 32 | Thread | 0h | RW | **TURBO_DISABLE —** 1 = Turbo Mode disabled, 0=Turbo Mode enabled. Note: This Turbo Mode Disable exists on each logical processor. Turbo Mode setting of a core does not impact Turbo Mode of the package.<br>BIOS should not use this Turbo Mode Disable since it does not change the Turbo Mode feature flag. This Turbo Mode Disable is intended for OS use only. |
| 31:16 | - | - | - | Reserved. |
| 15:8 | Thread | 0h | RW | **P_REQ —** Requested P-state ratio value for this logical processor. |
| 7:0 | - | - | - | Reserved. |

## 69.2.49 (19Ah) IA32_CLOCK_MODULATION

Thread T-State Request

| MSR Address: 19Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:5 | - | - | - | **Reserved (RSVD) —** Reserved. |
| 4 | Thread | 0h | RW | **THROTTLE_ACTIVE —** Enables ondemand software controlled clock modulation when set disables softwarecontrolled clock modulation when clear. |
| 3:0 | Thread | 0h | RW | **T_STATE_REQ —** Selects the ondemand clock modulation duty cycle. This field is only active when the ondemand clock modulation enable flag is set. |

## 69.2.50 (19Bh) IA32_THERM_INTERRUPT

Intel® Thermal Monitor Interrupt Control Register for the digital thermometer. When any of the interrupt enables are set, SW must configure the interrupt type to be delivered. SW writes the Thermal Local Vector Table (TLVT) entry of the local APIC to specify the interrupt type to be delivered.

| MSR Address: 19Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:25 | - | - | - | Reserved. |
| 24 | Core | 0h | RW | **SPARE_PWR_LIMIT_ENABLE —** Reserved for Power Limit Notification Enable. |
| 23 | Core | 0h | RW | **THRESHOLD_2_INT_ENABLE —** Threshold 2 Interrupt Enable R/W. This bit allows the BIOS to enable the generation of an interrupt on the transition of threshold 2 in any direction. A value 0 default disables interrupts and a value 1 enables interrupts.. |
| 22:16 | Core | 0h | RW | **THRESHOLD_2_REL_TEMP —** Specifies temperature Threshold 2 value as an offset below $TJ_{MAX}$ temperature. Compares against temperature reported value in the status register IA32_CR_THERM_STATUSDIG_TEMP_READOUT value. |
| 15 | Core | 0h | RW | **THRESHOLD_1_INT_ENABLE —** Threshold 1 Interrupt Enable R/W. This bit allows the BIOS to enable the generation of an interrupt on the transition of threshold 1 in any direction. A value 0 default disables interrupts and a value 1 enables interrupts. |
| 14:8 | Core | 0h | RW | **THRESHOLD_1_REL_TEMP —** Specifies temperature Threshold 1 value as an offset below $TJ_{MAX}$ temperature. Compares against temperature reported value in the status register IA32_CR_THERM_STATUSDIG_TEMP_READOUT value. |
| 7:5 | - | - | - | Reserved. |
| 4 | Core | 0h | RW | **OUT_OF_SPEC_INT_ENABLE —** This bit allows the BIOS to enable the generation of an interrupt on the transition to the OutofSpec condition. The OutofSpec temperature is specified as an offset from $TJ_{MAX}$ in CPU_THERM_CFG1OUT_OF_SPEC_THRESHOLD..If OVERHEAT_OFF_FUSE will ignore OOS condition. |
| 3 | - | - | - | Reserved. |
| 2 | Core | 0h | RW | **PROCHOT_INT_ENABLE —** This bit allows the BIOS to enable the interrupt on the transition of the external PROCHOT pin. If set a thermal interrupt is generated on either the rising or falling edge of PROCHOT.0 disable interrupt. 1 enable interrupt. |
| 1 | Core | 0h | RW | **LOW_TEMP_INT_ENABLE —** This bit allows the BIOS to enable the generation of an interrupt on the transition from hightemperature to a lowtemperature where high temperature is dictated by the thermal monitor trip threshold. 0 disable interrupt. 1 enable interrupt |
| 0 | Core | 0h | RW | **HIGH_TEMP_INT_ENABLE —** This bit allows the BIOS to enable the generation of an interrupt on the transition from lowtemperature to a hightemperature threshold where high temperature is dictated by the thermal monitor trip threshold. 0 disable interrupt. 1 enable interrupt. |

## 69.2.51 (19Ch) IA32_THERM_STATUS

Intel® Thermal Monitor CPU Core thermal status register.

| MSR Address: 19Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Core | 0h | RO | **VALID —** Set if temperature is within the valid thermal sensor range. |
| 30:27 | Core | 1h | RO | **RESOLUTION —** Supported resolution in degrees C. |
| 26:23 | - | - | - | Reserved. |
| 22:16 | Core | 0h | RO | **TEMPERATURE —** This is a temperature offset in degrees C from the $TJ_{MAX}$ temperature. See the READING_VALID field in the thermal status register for further details on how to interpret this field. Note that behavior of this field is also affected by CPU_THERM_SENS_CFGGROUP_TEMP. The power controller will read sensor data from each CPU core and update this field. |
| 15:12 | - | - | - | Reserved. |
| 11 | Core | 0h | RW_0C | **POWER_LIMITATION_LOG —** PWR_LIMIT Log. This bit is set by firmware and cleared by software. |
| 10 | Core | 0h | RO | **POWER_LIMITATION_STATUS —** PWR_LIMIT Status. |
| 9 | Core | 0h | RW_0C | **THRESHOLD2_LOG —** Sticky log bit that asserts on a 0 to 1 or a 1 to 0 transition of the THERMAL_THRESHOLD2_STATUS bit. This bit is set by firmware and cleared by SW. |
| 8 | Core | 0h | RO | **THRESHOLD2_STATUS —** Indicates that the current temperature reported in DIG_TEMP_READOUT is higher than or equal to Threshold 2 temperature. |
| 7 | Core | 0h | RW_0C | **THRESHOLD1_LOG —** Sticky log bit that asserts on a 0 to 1 or a 1 to 0 transition of the THERMAL_THRESHOLD1_STATUS bit. This bit is set by firmware and cleared by SW. |
| 6 | Core | 0h | RO | **THRESHOLD1_STATUS —** Indicates that the current temperature reported in DIG_TEMP_READOUT is higher than or equal to Threshold 1 temperature. |
| 5 | Core | 0h | RW_0C | **OUT_OF_SPEC_LOG —** Sticky log bit indicating that the processor operating out of its thermal specification since the last time this bit was cleared. This bit is set by firmware on a 0 to 1 transition of OUT_OF_SPEC_STATUS. |
| 4 | Core | 0h | RO | **OUT_OF_SPEC_STATUS —** Status bit indicating that the processor is operating out of its thermal specification. Once set this bit will be cleared once the temperature decreases back to cold condition. |
| 3 | Core | 0h | RW_0C | **PROCHOT_LOG —** Sticky log bit indicating that PROCHOT has been asserted since the last time this bit was cleared by SW. This bit is set by firmware on a 0 to 1 transition of PROCHOT_EVENT. |
| 2 | Core | 0h | RO | **PROCHOT_STATUS —** Status bit indicating that PROCHOT is currently being asserted. |
| 1 | Core | 0h | RW_0C | **THERMAL_MONITOR_LOG —** Sticky log bit indicating that the core has seen a thermal monitor event since the last time SW cleared this bit. This bit is set by Firmware on a 0 to 1 transition of THERMAL_MONITOR_STATUS. |
| 0 | Core | 0h | RO | **THERMAL_MONITOR_STATUS —** Status bit indicating that the Thermal Monitor has tripped and is currently thermally throttling. |

## 69.2.52 (19Dh) MSR_THERM2_CTL

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 19Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | - | - | - | Reserved. |
| 31:0 | Thread | 0h | Rsvd0 | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

## 69.2.53 (1A0h) IA32_MISC_ENABLE

Miscellaneous Enables Register holds enable bits for miscellaneous features.

| MSR Address: 1A0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:39 | - | - | - | Reserved. |
| 38 | Package | 0h | RW | **TURBO_DISABLE —** Disables turbo mode. |
| 37:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | RW | **NX_DISABLE —** When set to a 1, disables the Execute Disable Bit feature (XD Bit). |
| 33:24 | - | - | - | Reserved. |
| 23 | Thread | 0h | RW | **DISABLE_ECHOING_TPR_MESSAGES —** This MSR field maps to PIC_CR_DEBUG_MODES.TPR_MSG_OFF |
| 22 | Thread | 0h | RW | **BOOT_NT4 —** Setting this bit causes CPUID to pretend leaves 4 and 5 don't exist and that the max defined leaf is 3. |
| 21:19 | - | - | - | Reserved. |
| 18 | Thread | 0h | RW | **ENABLE_MONITOR_FSM —** Reset sets this bit. SW Enables MONITOR/MWAIT FSM in BLS. |
| 17 | - | - | - | Reserved. |
| 16 | Package | 0h | RW | **ENABLE_GV3 —** Setting this bit enables the Geyserville3 mechanism. The software can set the processor operating point by writing the proper VID and frequency ratio into the CLK_GEYSIII_CONTROL register. Default value is 0 (disabled). This bit is affected by fuse settings, and can be set to 1 only if FB_GV3_DISABLE_FUSE is '0 . This bit is write protected by the GV_SEL_LOCK bit. |
| 15:13 | - | - | - | Reserved. |
| 12 | Core | 0h | RO | **PEBS_UNAVAILABLE —** Based Sampling debug feature. RDMSR ucode returns !EMON_AVAILABLE. Written value is ignored. |
| 11 | Core | 0h | Rsvd0 | **BTS_UNAVAILABLE —** If set the processor does not support Branch Trace Stores. WRMSR prevents setting this bit. |
| 10:8 | - | - | - | Reserved. |
| 7 | Core | 0h | RW | **EMON_AVAILABLE —** RDMSR gets this bit from PM_CR_DEBUG_DEFEATURE.PERFMON_AVAIL. WRMSR masks writes to this bit. This bit prevents public software access to the emon / performance counter registers. When set, these regs are available and read/write. When clear, the registers. |
| 6:4 | - | - | - | Reserved. |
| 3 | Package | 0h | RW | **THERMAL_MONITOR_ENABLE —** This value from PCU_CR_FIRM_CONFIG. Global TM EN. This is a union of bits 3, 13 (ENABLE_TT2) and 36 (ENABLE_EMTTM). A value = 0 (default) indicates disabled, and a value = 1 indicates enabled. Activation is possible only if FB_THERM_THROT_DIS_FUSE is '0'". |
| 2 | - | - | - | Reserved. |
| 1 | - | - | - | Reserved. |
| 0 | Thread | 0h | RW | **FAST_STRINGS —** Enable fast strings. |

## 69.2.54 (1A1h) PACKAGE_THERM_MARGINBD

This register can be read to get the average Thermal margin above Tspec.

| MSR Address: 1A1h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:16 | - | - | - | Reserved. |
| 15:0 | Package | 0h | RO | **THERM_MARGIN —** Thermal margin in 8.8 format<br>. |

## 69.2.55 (1A2h) MSR_TEMPERATURE_TARGET

Legacy register holding temperature related constants for Platform use. This register is updated by FW based on fuse values.

| MSR Address: 1A2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | - | - | - | Reserved. |
| 30:24 | Package | 0h | RW | **TJ_MAX_TCC_OFFSET —** This field allows platform SW to configure a Tjmax_offset such that thermal throttling is activated at fused_Tjmax Tjmax_TCC_offset. FW reads this field to control thermal throttling activation at temperature lower than fused Tjmax. |
| 23:16 | Package | 5Ah | RO | **REF_TEMP —** Tjmax a.k.a. Throttle Temperature TCC activation temperature or Prochot Temperature or REF_TEMP This is the maximum junction temperature at which thermal throttling aka thermal monitor is activated. FW updates this register based on FUSE_TJMAX. This field is read only by all other SW. |
| 15:8 | Package | 0h | RO | **FAN_TEMP_TARGET_OFFSET —** Fan Temperature Target Offset a.k.a. TControl indicates the relative offset from the Thermal Monitor Trip Temperature at which fans should be engaged. FUSE_TEMP_TARGET 1. |
| 7:0 | - | - | - | Reserved. |

## 69.2.56 (1A4h) MSR_MISC_FEATURE_CONTROL

Miscellaneous Feature Control Register.

| MSR Address: 1A4h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:3 | - | - | - | Reserved. |
| 2 | Core | 0h | RW | **DCU_STREAMER_PREFETCH_DISABLE —** WMRSR maps this to MEC_CR_MODES_AND_DEFEATURES.DISABLE_PREFETCHER. |
| 1 | - | - | - | Reserved. |
| 0 | Module | 0h | RW | **MLC_STREAMER_PREFETCH_DISABLE —** WMRSR maps this to EBL_CR_DEBUG_CTL1_PREF_UCODE_DISABLE. |

## 69.2.57 (1A6h) MSR_OFFCORE_RSP_0

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 1A6h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | - | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

## 69.2.58 (1A7h) MSR_OFFCORE_RSP_1

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 1A7h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | - | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR. |

## 69.2.59 (1A8h) THREAD_SW_DEFEATURE

This MSR has no functionality and exists only for compatibility purposes. Returns 0s on reads, drops writes.

| MSR Address: 1A8h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:0 | Thread | 0h | RW | **DATA —** Returns zeros for RDMSR. Writes are dropped for WRMSR. |

## 69.2.60 (1AAh) MSR_MISC_PWR_MGMT

Legacy Miscellaneous Power Management Features.

| MSR Address: 1AAh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:23 | - | - | - | Reserved. |
| 22 | Package | 0h | RW | **LOCK_THERM_INT —** Ties thermal interrupts from both cores. If set then thermal interrupt on one core is routed to all cores. |
| 21:1 | - | - | - | Reserved. |
| 0 | Package | 0h | RW | **SINGLE_PCTL_EN —** When set (1) will cause PST commands (writes to IA32_PERF_CTL) from any cores to set the P-state target (no coordination). This means that the last that write the P-state request, is the dominant one. If not set this means that coordination between all the thread requests will be initiated and the resolved result will dominant. Default = 0 (disabled). Indicates the number of error-reporting banks available on this processor. |

## 69.2.61 (1ADh) MSR_TURBO_RATIO_LIMIT

PUNIT_MSR: This register allows determines the ratio limits to be used to limit frequency based on cores being C6/C7. During reset the SoC initializes this register based on SoC fuses. The MSR values determine how many ratio ticks of turbo to allow based on the number of active cores.

| MSR Address: 1ADh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:56 | Package | 0h | RW | **RATIO_LIMIT_7 —** Ratio limit 7 |
| 55:48 | Package | 0h | RW | **RATIO_LIMIT_6 —** Ratio limit 6 |
| 47:40 | Package | 0h | RW | **RATIO_LIMIT_5 —** Ratio limit 5 |
| 39:32 | Package | 0h | RW | **RATIO_LIMIT_4 —** Ratio limit 4 |
| 31:24 | Package | 0h | RW | **RATIO_LIMIT_3 —** Ratio limit 3 |
| 23:16 | Package | 0h | RW | **RATIO_LIMIT_2 —** Ratio limit 2 |
| 15:8 | Package | 0h | RW | **RATIO_LIMIT_1 —** Ratio limit 1 |
| 7:0 | Package | 0h | RW | **RATIO_LIMIT_0 —** Ratio limit 0 |

## 69.2.62 (1AEh) MSR_TURBO_RATIO_LIMIT1

PUNIT_MSR: This register allows determines the core counts to be used to limit frequency based on cores being C6/C7. During reset the SoC initializes this register based on SoC fuses. The MSR values determine how many ratio ticks of turbo to allow based on the number of active cores.

| MSR Address: 1AEh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:56 | Package | 0h | RW | **CORE_COUNT_7 —** Core count 7 |
| 55:48 | Package | 0h | RW | **CORE_COUNT_6 —** Core count 6 |
| 47:40 | Package | 0h | RW | **CORE_COUNT_5 —** Core count 5 |
| 39:32 | Package | 0h | RW | **CORE_COUNT_4 —** Core count 4 |
| 31:24 | Package | 0h | RW | **CORE_COUNT_3 —** Core count 3 |
| 23:16 | Package | 0h | RW | **CORE_COUNT_2 —** Core count 2 |
| 15:8 | Package | 0h | RW | **CORE_COUNT_1 —** Core count 1 |
| 7:0 | Package | 0h | RW | **CORE_COUNT_0 —** Core count 0 |

## 69.2.63 (1B0h) IA32_ENERGY_PERF_BIAS

Not a real register. This define exists to create the MSR routine that will be used to control the energy-efficient P-State performance policy. The bits will be redirected to a corresponding field in THREAD_P_REQ. This register allows software to control the Energy efficiency Policy in the CPU. This MSR will #GP unless POWER_CTL1[PWR_PERF_PLTFRM_OVR] is set.

| MSR Address: 1B0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:4 | - | - | - | Reserved. |
| 3:0 | Package | 0h | RW | **ENERGY_POLICY —** If enabled via POWER_CTL1, overrides the energy-efficient P-State performance policy field in THREAD_P_REQ |

## 69.2.64 (1B1h) IA32_PACKAGE_THERM_STATUS

Package Thermal Status Register for tracking package level temperature changes. This MSR is visible and accessible only when CPUID.(EAX=06h):EAX[6] is set to indicate this feature is available.

| MSR Address: 1B1h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Package | 0h | RW | **VALID —** Set if temperature is within the valid thermal sensor range. |
| 30:27 | Package | 1h | RO | **RESOLUTION —** Supported resolution in degrees C. |
| 26:23 | - | - | - | Reserved. |
| 22:16 | Package | 0h | RO | **TEMPERATURE —** Temperature in degrees C relative to the thermal monitor trip temperature fused. |
| 15:12 | - | - | - | Reserved. |
| 11 | Package | 0h | RW_0C | **POWER_LIMITATION_LOG —** Sticky log bit that asserts when either IA is running at P-state below the max P-state offset or that GT is running at P-state below its P1 frequency. Set by HW on a 0 to 1 transition of Power Limitation Status. Cleared by SW. |
| 10 | Package | 0h | RO | **POWER_LIMITATION_STATUS —** Status log bit that notifies that either IA is running at P-state below the max P-state offset or that GT is running at P-state below its P1 frequency. |
| 9 | Package | 0h | RW_0C | **THRESHOLD2_LOG —** Sticky log bit that asserts on a 0 to 1 or 1 to 0 transition of the Threshold1_Status bit. HW controls this transition. |
| 8 | Package | 0h | RO | **THRESHOLD2_STATUS —** Indicates that the current temperature bits 22:16 in this register is higher than or equal to the Threshold1 defined in the IA32_PACKAGE_THERM_INTERRUPT MSR. Note that because temperature and thresholds are defined as negative offsets a higher number means a lower temperature. |
| 7 | Package | 0h | RW_0C | **THRESHOLD1_LOG —** Sticky log bit that asserts on a 0 to 1 or 1 to 0 transition of the Threshold1_Status bit. HW controls this transition. |
| 6 | Package | 0h | RO | **THRESHOLD1_STATUS —** Indicates that the current temperature bits 22:16 in this register is higher than or equal to the Threshold1 defined in the IA32_PACKAGE_THERM_INTERRUPT MSR. Note that because temperature and thresholds are defined as negative offsets a higher number means a lower temperature. |
| 5 | Package | 0h | RW_0C | **OUT_OF_SPEC_LOG —** Sticky log bit indicating that the processor operating out of its thermal specification since the last time this bit was cleared. Set by HW on a 0 to 1 transition of Out_of_Spec_Status |
| 4 | Package | 0h | RO | **OUT_OF_SPEC_STATUS —** Status bit indicating that the processor is operating out of its thermal specification. |
| 3 | Package | 0h | RW_0C | **PROCHOT_LOG —** Sticky log bit indicating that PROCHOT has been asserted since the last time this bit was cleared by SW. Set by HW on a 0 to 1 transition of Prochot_Status. |
| 2 | Package | 0h | RO | **PROCHOT_STATUS —** Status bit indicating that PROCHOT is currently being asserted. |
| 1 | Package | 0h | RW_0C | **THERMAL_MONITOR_LOG —** Sticky log bit indicating that the package has seen a thermal monitor event since the last time SW cleared this bit. Set by HW on a 0 to 1 transition of Thermal_Monitor_Status. |
| 0 | Package | 0h | RO | **THERMAL_MONITOR_STATUS —** Status bit indicating that any of the package thermal monitor have tripped and the package is currently thermally throttling. |

## 69.2.65 (1B2h) IA32_PACKAGE_THERM_INTERRUPT

Thermal Limitation Interrupt Control. PCODE will read this information before generating a thermal interrupt.

| MSR Address: 1B2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:25 | - | - | - | Reserved. |
| 24 | Package | 0h | RW | **POWER_INT_ENABLE —** When this bit is set a thermal interrupt will be sent upon throttling due to power limitations. |
| 23 | Package | 0h | RW | **THRESHOLD_2_INT_ENABLE —** Controls the generation of a thermal interrupt whenever the Thermal Threshold 2 Temperature is crossed. |
| 22:16 | Package | 0h | RW | **THRESHOLD_2_REL_TEMP —** This value indicates the offset in degrees below TJ Max Temperature that should trigger a Thermal Threshold 2 trip. |
| 15 | Package | 0h | RW | **THRESHOLD_1_INT_ENABLE —** Controls the generation of a thermal interrupt whenever the Thermal Threshold 1 Temperature is crossed. |
| 14:8 | Package | 0h | RW | **THRESHOLD_1_REL_TEMP —** This value indicates the offset in degrees below TJ Max Temperature that should trigger a Thermal Threshold 1 trip. |
| 7:5 | - | - | - | Reserved. |
| 4 | Package | 0h | RW | **OUT_OF_SPEC_INT_ENABLE —** Thermal interrupt enable for the Out Of Spec condition which is stored in the Out Of Spec status bit in PACKAGE_THERM_STATUS. |
| 3 | - | - | - | Reserved. |
| 2 | Package | 0h | RW | **PROCHOT_INT_ENABLE —** Bidirectional PROCHOT assertion interrupt enable. If set a thermal interrupt is delivered on the rising edge of PROCHOT. |
| 1 | Package | 0h | RW | **LOW_TEMP_INT_ENABLE —** Enables a thermal interrupt to be generated on the transition from a hightemperature to a lowtemperature when set where high temperature is dicated by the thermal monitor trip temperature minus offset as defined in IA32_TEMPERATURE_TARGET. |
| 0 | Package | 0h | RW | **HIGH_TEMP_INT_ENABLE —** Enables a thermal interrupt to be generated on the transition from a lowtemperature to a hightemperature when set where high temperature is dicated by the thermal monitor trip temperature minus offset as defined in IA32_TEMPERATURE_TARGET. |

## 69.2.66 (1C6h) DEBUG_RESOURCE_STATUS

This MSR returns the status of debug resources.

| MSR Address: 1C6h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | RO | **DEBUG_AVAILABLE —** |
| 0 | Thread | 0h | RO | **DEBUG_DRX_AVAILABLE —** |

## 69.2.67 (1C8h) MSR_LBR_SELECT

MSR_LBR_SELECT is cleared to zero at RESET, and LBR (Last Branch Record) filtering is disabled, i.e. all branches will be captured. MSR_LBR_SELECT provides bit fields to specify the conditions of subsets of branches that will not be captured in the LBR.

| MSR Address: 1C8h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:10 | - | - | - | Reserved. |
| 9 | Thread | 0h | RW | **LBR_CALL_STACK_EN —** Enable LBR call stack. |
| 8 | Thread | 0h | RW | **FAR_BRANCH —** Filter out far branches. |
| 7 | Thread | 0h | RW | **NEAR_REL_JMP —** Filter out near unconditional relative branches. |
| 6 | Thread | 0h | RW | **NEAR_INDIRECT_JMP —** Filter out near unconditional indirect jumps. |
| 5 | Thread | 0h | RW | **NEAR_RET —** Filter out near returns. |
| 4 | Thread | 0h | RW | **NEAR_INDIRECT_CALL —** Filter out near indirect calls. |
| 3 | Thread | 0h | RW | **NEAR_REL_CALL —** Filter out near relative calls. |
| 2 | Thread | 0h | RW | **JCC —** Filter out taken conditional branches. |
| 1 | Thread | 0h | RW | **CPL_NEQ_0 —** Filter out non-ring 0 branches. |
| 0 | Thread | 0h | RW | **CPL_EQ_0 —** Filter out ring 0 branches. |

## 69.2.68 (1C9h) MSR_LASTBRANCH_TOS

Contains an index (bits 0-3) that points to the MSR containing the most recent branch record. See MSR_LASTBRANCH_0_FROM_IP (at 0x680).

| MSR Address: 1C9h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:5 | - | - | - | Reserved. |
| 4:0 | Thread | 0h | RW | **TOS_POINTER —** LBR Top-of-stack. |

## 69.2.69 (1D9h) IA32_DEBUGCTL

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: 1D9h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:15 | - | - | - | Reserved. |
| 14 | Thread | 0h | RW | **FREEZE_WHILE_SMM_EN —** When set, freezes perfmon and trace messages while in SMM. When set, freezes perfmon and trace messages while in SMM. |
| 13 | - | - | - | Reserved. |
| 12 | Thread | 0h | RW | **FREEZE_PERFMON_ON_PMI —** When set, each ENABLE bit of the global counter control MSR are frozen (address 3BFH) on a PMI request. |
| 11 | Thread | 0h | RW | **FREEZE_LBRS_ON_PMI —** When set, the LBR stack is frozen on a PMIrequest. |
| 10 | Thread | 0h | RW | **BTS_OFF_USR —** When set, BTS or BTM is skipped if CPL > 0 |
| 9 | Thread | 0h | RW | **BTS_OFF_OS —** When set, BTS or BTM is skipped if CPL = 0. |
| 8 | Thread | 0h | RW | **BTINT —** When clear, BTMs are logged in a BTS buffer in circular fashion. When this bit is set, an interrupt is generated by the BTS facility when the BTS buffer is full. |
| 7 | Thread | 0h | RW | **BTS —** Setting this bit enables branch trace messages (BTMs) to be logged in a BTS buffer. |
| 6 | Thread | 0h | RW | **TR —** Setting this bit to 1 enables branch trace messages to be sent. |
| 5:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | RW | **BTF —** Setting this bit to 1 enables the processor to treat EFLAGS.TF as single-step on branches instead of single-step on instructions. |
| 0 | Thread | 0h | RW | **LBR —** Setting this bit to 1 enables the processor to record a running trace of the most recent branches taken by the processor in the LBR stack. |

## 69.2.70 (1E3h) DEBUG_ERR_INJ_CTL

This MSR holds the capability of MCA error injection. Bit 1 for unlock MCA/ CMCI generation capability (allows event injection), bit 0 to unlock writing to MC banks. This register contains controls to enable/disable both the MB bank write capability and the MCA/CMCI signaling capability.

| MSR Address: 1E3h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | RW | **MCBW_E —** CMCI/MCA signaling enable flag. If this bit is set, enables MCA and CMCI signaling through DEBUG_ERR_INJ_CTL2 register. |
| 0 | Thread | 0h | RW | **MCA_CMCI_SE —** Machine check bank write capability enable flag - If this bit is set, non-zero value writes to the MCi_STATUS, MCi_MISC and MCi_ADDR registers are allowed (will not cause general-protection (#GP) exception) from SMM and ring0. If this bit is clear, regular MC bank access rules apply to the MCi_STATUS, MCi_MISC and MCi_ADDR registers (i.e, cause general-protection (#GP) exception on non-zero value writes). |

## 69.2.71 (1E4h) DEBUG_ERR_INJ_CTL2

This register allows generating MCA or CMCI from ring0 code. This is write only (WO) register, it can be written only if DEBUG_ERR_INJ_CTL.MCA_CMCI_SE bit is set. Writing to this MSR without setting MCA_CMCI_SE bit will cause general-protection (#GP) exception. Reads to this MSR will cause general-protection (#GP) exception.

| MSR Address: 1E4h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | WO | **MCA_G —** Setting this bit will cause MCA to be broadcast to all threads in the system. |
| 0 | Thread | 0h | WO | **CMCI_G —** Setting this bit causes a CMCI to be generated on the both cores on the module when this MSR is written. |

***Note:*** Specifies the machine-check architecture defined error code for the machine check error condition detected. The machine-check architecture-defined error codes are guaranteed to be the same for all IA-32 processors that implement the machine-check architecture. Please refer to section 15.9 and chapter 16 of the document 325384 “Intel 64 and IA-32 Intel Architecture Software Developer's manual Volume 3: system programming guide” for the detail information.

## 69.2.72 (1E6h) PIC_DEBUG_MODES

| MSR Address: 1E6h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Module | 0h | RW | **SELF_IPI_OFF —** CB to disable the Self IPI feature and revert Self IPI interrupt behavior. |
| 30 | Module | 0h | RW | **DISABLE_CTC_DL_SYNC —** CB to disable CTC sync with URT using the sync (Q) bit. |
| 29 | - | - | - | Reserved. |
| 28 | Module | 0h | RW | **WATCHDOG_IGNORE_BUSQEMPTY —** Masks out bus queue empty from smc nuke logic. |
| 27 | Module | 0h | RW | **WATCHDOG_QUICK —** Change all the above WATCHDOG setting to be on ROB Timeout[1] instead of ROB Timeout[2], cut timeout times in half to give a warning early to the other core. |
| 26 | Module | 0h | RW | **DISABLE_ASYNC_MCKIND —** Disable Asynchronous MCKind from being signaled |
| 25 | Module | 0h | RW | **DISABLE_ASYNC_MCERR —** Disable Asynchronous MCERR from being signaled |
| 24 | Module | 0h | RW | **PSMI_ASSIST —** Prevents certain registers from changing and prevents side-effects for PSMI Modes |
| 23 | Module | 0h | RW | **WATCHDOG_SMC_OTHER_REPEATED —** On ROB Timeout[2] = 1, SMC Nuke other core continuously |
| 22 | Module | 0h | RW | **WATCHDOG_SMC_SAME_WAIT —** On ROB Timeout Cnt 011 to 100, SMC Nuke same core on TSC[8] Toggle |
| 21 | Module | 0h | RW | **WATCHDOG_SMC_SAME_NOW —** On ROB Timeout Cnt 011 to 100, SMC Nuke same core now |
| 20 | Module | 0h | RW | **WATCHDOG_SMC_OTHER_WAIT —** On ROB Timeout Cnt 011 to 100, SMC Nuke other core on TSC[8] Toggle |
| 19 | Module | 0h | RW | **WATCHDOG_SMC_OTHER_NOW —** On ROB Timeout Cnt 011 to 100, SMC Nuke other core now |
| 18:17 | Module | 0h | RW | **PERIODIC_SMC_TSC_SEL —** Selects Bit of TSC for Periodic SMC Nuke. 00:TSC[20]:1ms, 00:TSC[20]:1ms, 01:TSC[23]:10ms, 10:TSC[27]:100ms, 11:TSC[30]:1s |
| 16 | Module | 0h | RW | **PERIODIC_SMC_EN —** Enables Periodic SMC Nukes if set |
| 15 | Module | 0h | RW | **CMCI_FEATURE_DISABLE —** Setting this bit will disable the CMCI LVT |
| 14 | Module | 0h | RW | **L2_CMCI_INTERNAL —** Setting this bit will not send the L2 CMCI event to Tunit and complete internally instead. |
| 13 | - | - | - | Reserved. |
| 12 | Module | 0h | RW | **ID_IS_NOT_READONLY —** when set to '1 enables SW to write to ID register in APIC. |
| 11 | Module | 0h | RW | **TSC_DEADLINE_CB —** CB for TSC Deadline |
| 10 | Module | 0h | RW | **SV_SENTER_EN —** enables saving of internal data to SCPs during SENTER |
| 9 | Module | 0h | RW | **SV_GLOBAL_ENABLE —** enables PATCH2 and PATCH3 special instructions in Unlocked units only. |
| 8 | Module | 0h | RW | **DISABLE_INTPRIUPD_GENERATION —** >Disables TPR/IntPriUpd from being generated |
| 7 | Module | 0h | RW | **EOI_NO_BDCST_ENABLE —** f |
| 6 | Module | 0h | RW | **EOI_SUPPRESS_CB —** |
| 5 | Module | 0h | RW | **ICR_MSG_OFF —** |
| 4 | Module | 0h | RW | **TPR_MSG_OFF —** Disable new TPR Update message requests to the FSB TPR messages. Will be generated for changes to PIC HW/SW enables. |

| MSR Address: 1E6h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 3 | Module | 0h | RW | **PIC_REQ_OFF —** Disable new PICOMB requests to the EBC. |
| 2 | Module | 0h | RW | **CCR_FREEZE_CORE1 —** When this bit is set the CCR of Core1 will be frozen |
| 1 | Module | 0h | RW | **CCR_FREEZE_CORE0 —** When this bit is set the CCR of Core0 will be frozen |
| 0 | Module | 0h | RW | **TSC_FREEZE —** When this bit is set its freezes the counting of the Time Stamp counter.) |

## 69.2.73 (1F0h) VLW_CAPABILITY

This MSR reports information about the Legacy features and VLW messages that are supported.

| MSR Address: 1F0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:7 | - | - | - | Reserved. |
| 6 | Core | 0h | RO | **NMI —** 1 means NMI VLW message is supported<br>1 = Supported, 0= Not Supported. |
| 5 | Core | 0h | RO | **INIT —** 1 means INIT VLW message is supported<br>1 = Supported, 0= Not Supported. |
| 4 | Core | 0h | RO | **SMI —** 1 means SMI VLW message is supported. This will be 0 if SMM is fuse disabled |
| 3 | - | - | - | Reserved. |
| 2 | Core | 0h | RO | **INTR —** Indicates whether INTR VLW message is supported.<br>1 = Supported, 0= Not Supported. |
| 1:0 | - | - | - | Reserved. |

## 69.2.74 (1FCh) MSR_POWER_CTL

Power Control Register allows enable and disable of specific power and thermal management features.

| MSR Address: 1FCh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | - | - | - | Reserved. |
| 30 | Package | 0h | RW | **CSTATE_PREWAKE_DISABLE —** This bit can be used to disable core/pkg C-state Prewake timers. |
| 29:25 | - | - | - | Reserved. |
| 24 | Package | 0h | RW | **VR_THERM_ALERT_DISABLE —** When set to 1, disables the VR_THERMAL_ALERT signaling. |
| 23 | Package | 0h | RW | **PROCHOT_LOCK —** When set to 1, locks bits 0, 21, 22 and 23 of this MSR. Once set, a reset is required to clear this bit. |
| 22 | Package | 0h | RW | **PROCHOT_RESPONSE —** Prochhot Configurable Response Enable. |
| 21 | Package | 0h | RW | **PROCHOT_OUTPUT_MODE_DISABLE —** Prochot output disable. |
| 20 | - | - | - | Reserved. |
| 19 | Package | 0h | RW | **EE_TURBO_DISABLE —** Energy Efficient Turbo Disable. |
| 18 | Package | 0h | RW | **PWR_PERF_PLTFRM_OVR —** When set to 0 (default), will disable access to IA32_ENERGY_PERFORMANCE_BIAS MSR 1B0h (causing a GP# fault on access). CPUID.(EAX=6):ECX[3] will read 0. |
| 17 | Package | 0h | RW | **PHOLD_SR_DISABLE —** If set to '1', CPU will also prevent the memory from going to SR during the CST_PREVENTION timeout; |
| 16:6 | Package | 0h | RW | **PHOLD_CST_PREVENTION_INIT —** Init value for the PHOLD_CST_PREVENTION_Timer. Values are between 0 and 1023 in 2^21*10nS resolution (matching 0 - ~21sec). |
| 5 | - | - | - | Reserved. |
| 4 | Package | 0h | RW | **FAST_BRK_INT_EN —** Controls the VID swing rate for Other_Intp_Wake events detected at the iMPH. Will override a "SLOW" sideband wake indication (for C7 or PCIe). |
| 3 | Package | 0h | RW | **FAST_BRK_SNP_EN —** Controls the VID swing rate for Other_Snp_Wake events detected at the iMPH. Will override a "SLOW" sideband wake indication primarily for PCIe traffic. |
| 2 | Package | 0h | RW | **SAPM_IMC_C2_POLICY —** Enables Self Refresh entry when the CPU is in package-C2 (popup) state. Should be disabled only if platform traffic cannot withstand even the SR exit latency while in C2. |
| 1 | Package | 0h | RW | **C1E_ENABLE —** Used to enable C1E behavior on every C1 entry (even if MWAIT C1E substate bit was not set or when using HLT).When this package-visible bit is set, it will cause CPU to switch to minimum GV point when all cores have entered C1. Frequency will switch immediately, followed by gradual voltage switching (GV3 style). This is a package visible bit stored in PCU_CR_FIRM_CONFIG[C1E_ENABLE] |
| 0 | Package | 0h | RW | **ENABLE_BIDIR_PROCHOT —** Used to enable or disable the response to PROCHOT# input. When set/enabled, platform can force CPU to throttle to a lower power condition such as Pn/Pm by asserting prochot#. When clear/disabled (default), CPU ignores the status of the prochot input signal. |

## 69.2.75 (281h) IA32_MC1_CTL2 (L2 Cache)

Bank 1 MCA Control Register 2.

| MSR Address: 281h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Module | - | RO | Reserved. |
| 30 | Module | 0h | RW | **CMCI_EN —** CMCI Enable |
| 29:15 | Module | - | RO | Reserved. |
| 14:0 | Module | 0h | RW | **CORRECTED_ERROR_COUNT_THRESHOLD —** Corrected Error Count Threshold. |

## 69.2.76 (282h) IA32_MC2_CTL2 (FEC)

MCA Address Control Register for CMCI.

| MSR Address: 282h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Core | - | RO | Reserved. |
| 30 | Core | 0h | RW | **CMCI_EN —** CMCI Enable. Software sets this bit to enable the generation of corrected machine check interrupts. |
| 29:15 | Core | - | RO | Reserved. |
| 14:0 | Core | 0h | RW | **CORRECTED_ERROR_COUNT_THRESHOLD —** The Threshold Value for Corrected Error Count. |

## 69.2.77 (283h) IA32_MC3_CTL2 (MEC)

| MSR Address: 283h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Module | - | RO | Reserved. |
| 30 | Module | 0h | Rsvd0 | **CMCI_EN —** Reserved. |
| 29:15 | Module | - | RO | Reserved. |
| 14:0 | Module | 0h | Rsvd0 | **CORRECTED_ERROR_COUNT_THRESHOLD —** Reserved. |

## 69.2.78 (284h) IA32_MC4_CTL2 (SA)

| MSR Address: 284h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | Package | - | RO | Reserved. |
| 34 | Package | 0h | RW | **MCE_CTL —** Controls MCE Remapping 0 MCE 1 MSMI. |
| 33 | Package | - | RO | Reserved. |
| 32 | Package | 0h | RW | **CMCI_CTL —** Controls CMCI Remapping 0 CMCI 1 CSMI. |
| 31 | Package | - | RO | Reserved. |
| 30 | Package | 0h | RW | **CMCI_EN —** Reserved. |
| 29:15 | Package | - | RO | Reserved. |
| 14:0 | Package | 0h | RW | **THRESHOLD —** Reserved. |

## 69.2.79 (285h) IA32_MC5_CTL2 (PMU)

| MSR Address: 285h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | Package | - | RO | Reserved. |
| 34 | Package | 0h | RW | **MCE_CTL —** Controls MCE Remapping 0 MCE 1 MSMI. |
| 33 | Package | - | RO | Reserved. |
| 32 | Package | 0h | RW | **CMCI_CTL —** Controls CMCI Remapping 0 CMCI 1 CSMI. |
| 31 | Package | - | RO | Reserved. |
| 30 | Package | 0h | RW | **CMCI_EN —** Reserved. |
| 29:15 | Package | - | RO | Reserved. |
| 14:0 | Package | 0h | RW | **THRESHOLD —** Reserved. |

## 69.2.80 (286h) IA32_MC6_CTL2 (SA)

| MSR Address: 286h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | Package | - | RO | Reserved. |
| 34 | Package | 0h | RW | **MCE_CTL —** Controls MCE Remapping 0 MCE 1 MSMI. |
| 33 | Package | - | RO | Reserved. |
| 32 | Package | 0h | RW | **CMCI_CTL —** Controls CMCI Remapping 0 CMCI 1 CSMI. |
| 31 | Package | - | RO | Reserved. |
| 30 | Package | 0h | RW | **CMCI_EN —** Reserved. |
| 29:15 | Package | - | RO | Reserved. |
| 14:0 | Package | 0h | RW | **THRESHOLD —** Reserved. |

## 69.2.81 (287h) IA32_MC7_CTL2 (MEM 0)

| MSR Address: 287h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | Package | - | RO | Reserved. |
| 34 | Package | 0h | RW | **MCE_CTL —** Controls MCE Remapping 0 MCE 1 MSMI. |
| 33 | Package | - | RO | Reserved. |
| 32 | Package | 0h | RW | **CMCI_CTL —** Controls CMCI Remapping 0 CMCI 1 CSMI. |
| 31 | Package | - | RO | Reserved. |
| 30 | Package | 0h | RW | **CMCI_EN —** Reserved. |
| 29:15 | Package | - | RO | Reserved. |
| 14:0 | Package | 0h | RW | **THRESHOLD —** Reserved. |

## 69.2.82 (288h) IA32_MC8_CTL2 (MEM 1)

| MSR Address: 288h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | Package | - | RO | Reserved. |
| 34 | Package | 0h | RW | **MCE_CTL —** Controls MCE Remapping 0 MCE 1 MSMI. |
| 33 | Package | - | RO | Reserved. |
| 32 | Package | 0h | RW | **CMCI_CTL —** Controls CMCI Remapping 0 CMCI 1 CSMI. |
| 31 | Package | - | RO | Reserved. |
| 30 | Package | 0h | RW | **CMCI_EN —** Reserved. |
| 29:15 | Package | - | RO | Reserved. |
| 14:0 | Package | 0h | RW | **THRESHOLD —** Reserved. |

## 69.2.83 (2E0h) NO_EVICT_MODE

| MSR Address: 2E0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Thread | 0h | RW | **NEM_RUN —** BIOS sets this bit when it's done filling the NEM region and ready for using cache as RAM. |
| 0 | Thread | 0h | RW | **NEM_SETUP —** BIOS sets this when it is ready to fill the NEM region, and then executes the requests to fill the NEM. |

## 69.2.84 (2FFh) IA32_MTRR_DEF_TYPE

The IA32_MTRR_DEF_TYPE MSR sets the default properties of the regions of physical memory that are not encompassed by MTRRs.

| MSR Address: 2FFh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:12 | - | - | - | Reserved. |
| 11 | Core | 0h | RW | **MTRR_ENABLE —** When set, enables all MTRRs. A value = 0 disables, and a value = 1 enables all MTRRs.<br>Note: Global MTRR Enable is often referred to as IA32_MTRR_DEF_TYPE.E or IA32_MTRR_DEF_TYPE.GE. |
| 10 | Core | 0h | RW | **FIXED_RANGE_ENABLE —** When set, enables the fixed range MTRRs. A value = 0 disables, and a value = 1 enables the Fixed MTRRs.<br>Note: Fixed MTRR Enable is often referred to as IA32_MTRR_DEF_TYPE.F or IA32_MTRR_DEF_TYPE.FE. |
| 9:3 | - | - | RO | Reserved. |
| 2:0 | Core | 0h | RW | **TYPE —** Indicates the default memory type used for those memory ranges that do not have a memory type specified in an MTRR.<br>Note: Default MTRR Memory Type is often referred to as IA32_MTRR_DEF_TYPE. |

### 69.2.85 (300h) MSR_SGXOWNER0

Register shared by all threads.

| MSR Address: 300h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RW | **DATA —** Data space allowed to be read/written by the SoC in byte granularity. |

### 69.2.86 (301h) MSR_SGXOWNER1

Register shared by all threads.

| MSR Address: 301h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RW | **DATA —** Data space allowed to be read/written by the SoC in byte granularity. |

### 69.2.87 (309h) IA32_FIXED_CTR0

IA32_FIXED_CTR0 contains a 48-bit fixed rate counter.

| MSR Address: 309h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | - | - | - | Reserved. |
| 47:0 | Thread | 0h | RW | **FIXED_COUNTER0_VALUE —** 48-bit fixed rate counter 0. |

### 69.2.88 (30Ah) IA32_FIXED_CTR1

IA32_FIXED_CTR1 contains a 48-bit fixed rate counter.

| MSR Address: 30Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | - | - | - | Reserved. |
| 47:0 | Thread | 0h | RW | **FIXED_COUNTER1_VALUE —** 48-bit fixed rate counter 1. |

### 69.2.89 (30Bh) IA32_FIXED_CTR2

IA32_FIXED_CTR2 contains a 48-bit fixed rate counter.

| MSR Address: 30Bh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | - | - | - | Reserved. |
| 47:0 | Thread | 0h | RW | **FIXED_COUNTER2_VALUE —** 48-bit fixed rate counter 2. |

## 69.2.90 (345h) IA32_PERF_CAPABILITIES

This register enumerates the existence of certain debug features in PerfMon.

| MSR Address: 345h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:14 | - | - | - | Reserved. |
| 13 | Core | 0h | RO | **FULLWIDTHCNTRS —** Full width of counter writable via IA32_A_PMCx. |
| 12 | Core | 0h | RO | **SMMFREEZE —** Freeze while SMM is supported |
| 11:8 | Core | 0h | RO | **PEBS_REC_FORMAT —** PEBS Record Format. |
| 7 | Core | 0h | RO | **PEBS_SAVE_ARCH_REGS —** When set, PEBS will save architectural register and state information according to the encoded value of the PEBSRecordFormat field. |
| 6 | Core | 0h | RO | **PEBS_TRAP —** PEBS Record Format. |
| 5:0 | Core | 0h | RO | **LBR_FORMAT —** Last Branch Record Format. 0=32-bit, 1=64-bit LIP, 2=64-bit EIP. |

## 69.2.91 (38Dh) IA32_FIXED_CTR_CTRL

This register provides enables for the fixed performance counters and enables for PMI. This register also provides the control for the counts to be for single thread or both threads in a core.

| MSR Address: 38Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:12 | - | - | - | Reserved. |
| 11 | Thread | 0h | RW | **CTR_2_PMI_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 10 | Thread | 0h | RW | **CTR_2_BOTH_THREAD_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 9:8 | Thread | 0h | RW | **CTR_2_ENABLE —** This bit field is the local enable for the Fixed counter 2. Disabling these bits will cause the counters to halt. Counter is enabled by setting CPL Level: 00 - Disable 01 - Ring 0 (OS) 10 - Ring 1,2,3 (User) 11 - All ring levels (OS or User) when CPL changes to one of the above the counter gets enabled. |
| 7 | Thread | 0h | RW | **CTR_1_PMI_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 6 | Thread | 0h | RW | **CTR_1_BOTH_THREAD_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 5:4 | Thread | 0h | RW | **CTR_1_ENABLE —** This bit field is the local enable for the Fixed counter 2. Disabling these bits will cause the counters to halt. Counter is enabled by setting CPL Level: 00 - Disable 01 - Ring 0 (OS) 10 - Ring 1,2,3 (User) 11 - All ring levels (OS or User) when CPL changes to one of the above the counter gets enabled. |
| 3 | Thread | 0h | RW | **CTR_0_PMI_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 2 | Thread | 0h | RW | **CTR_0_BOTH_THREAD_ENABLE —** When this bit is set fixed counter 0 will count instructions retired for both threads. |
| 1:0 | Thread | 0h | RW | **CTR_0_ENABLE —** This bit field is the local enable for the Fixed counter 2. Disabling these bits will cause the counters to halt. Counter is enabled by setting CPL Level: 00 - Disable 01 - Ring 0 (OS) 10 - Ring 1,2,3 (User) 11 - All ring levels (OS or User) when CPL changes to one of the above the counter gets enabled. |

## 69.2.92 (38Eh) IA32_PERF_GLOBAL_STATUS

Global Performance Counter Status.

| MSR Address: 38Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Thread | 0h | RO | **COND_CHG —** Status bits of this register has changed. |
| 62 | Thread | 0h | RO | **PEBS_BUFFER_OVERFLOW —** DS SAVE area Buffer overflow status. |
| 61 | - | - | - | Reserved. |
| 60 | Thread | 0h | RO | **ASCI —** Set if any of the perfmon counters are suppressed. |
| 59 | Thread | 0h | RO | **COUNTERS_FROZEN —** |
| 58 | Thread | 0h | RO | **LBRS_FROZEN —** |
| 57:56 | - | - | - | Reserved. |
| 55 | Thread | 0h | RO | **RTIT_TOPA_PMI —** RTIT Table of Physical Addresses PMI. Allows software to differentiate between legacy PMI sources. |
| 54:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | RO | **FIXED_PERFMON_COUNTER_2_OVERFLOW —** Overflow status of IA32_FIXED_CTR2 |
| 33 | Thread | 0h | RO | **FIXED_PERFMON_COUNTER_1_OVERFLOW —** Overflow status of IA32_FIXED_CTR1 |
| 32 | Thread | 0h | RO | **FIXED_PERFMON_COUNTER_0_OVERFLOW —** Overflow status of IA32_FIXED_CTR0 |
| 31:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | RO | **GP_PERFMON_COUNTER_3_OVERFLOW —** Overflow status of IA32_PMC3. |
| 2 | Thread | 0h | RO | **GP_PERFMON_COUNTER_2_OVERFLOW —** Overflow status of IA32_PMC2. |
| 1 | Thread | 0h | RO | **GP_PERFMON_COUNTER_1_OVERFLOW —** Overflow status of IA32_PMC1. |
| 0 | Thread | 0h | RO | **GP_PERFMON_COUNTER_0_OVERFLOW —** Overflow status of IA32_PMC0. |

## 69.2.93 (38Fh) IA32_PERF_GLOBAL_CTRL

Counter increments while the result of ANDing respective enable bit in this MSR with the corresponding OS or USR bits in the general-purpose or fixed counter control MSR is true.

| MSR Address: 38Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | RW | **ENABLE_FIXED_PERFMON_COUNTER_2 —** Global Enable for Fixed Counter 2. Both this enable and the local enable for the fixed counter needs to be asserted for the fixed counter to be counting. |
| 33 | Thread | 0h | RW | **ENABLE_FIXED_PERFMON_COUNTER_1 —** Global Enable for Fixed Counter 1. Both this enable and the local enable for the fixed counter needs to be asserted for the fixed counter to be counting. |
| 32 | Thread | 0h | RW | **ENABLE_FIXED_PERFMON_COUNTER_0 —** Global Enable for Fixed Counter 0. Both this enable and the local enable for the fixed counter needs to be asserted for the fixed counter to be counting. |
| 31:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | RW | **ENABLE_GP_PERFMON_COUNTER_3 —** Global Enable for PerfMon Counter 3. If either the global or local enable are not asserted, the counter will be disabled. The global enable allows for simultaneous enabling or disabling of all or some of the Core PerfMon counters. |
| 2 | Thread | 0h | RW | **ENABLE_GP_PERFMON_COUNTER_2 —** Global Enable for PerfMon Counter 2. If either the global or local enable are not asserted, the counter will be disabled. The global enable allows for simultaneous enabling or disabling of all or some of the Core PerfMon counters. |
| 1 | Thread | 0h | RW | **ENABLE_GP_PERFMON_COUNTER_1 —** Global Enable for PerfMon Counter 1. If either the global or local enable are not asserted, the counter will be disabled. The global enable allows for simultaneous enabling or disabling of all or some of the Core PerfMon counters. |
| 0 | Thread | 0h | RW | **ENABLE_GP_PERFMON_COUNTER_0 —** Global Enable for PerfMon Counter 0. If either the global or local enable are not asserted, the counter will be disabled. The global enable allows for simultaneous enabling or disabling of all or some of the Core PerfMon counters. |

## 69.2.94 (390h) IA32_PERF_GLOBAL_STATUS_RESET

Global Perfmon Overflow clear register.

| MSR Address: 390h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Thread | 0h | WO | **CLR_CONDCHG —** If set, clears the conditions changed in the status register. |
| 62 | Thread | 0h | WO | **CLEAR_PEBS_BUFFER_OVERFLOW —** Clears related bit in IA32_CR_PERF_GLOBAL_STATUS. |
| 61 | - | - | - | Reserved. |
| 60 | Thread | 0h | WO | **CLEAR_ASCI —** Clear ASCI in status register. |
| 59 | Thread | 0h | WO | **CLEAR_COUNTERS_FROZEN —** Clears IA32_PERF_GLOBAL_STATUS[59]. |
| 58 | Thread | 0h | WO | **CLEAR_LBRS_FROZEN —** Clears IA32_PERF_GLOBAL_STATUS[58]. |
| 57:56 | - | - | - | Reserved. |
| 55 | Thread | 0h | WO | **CLEAR_RTIT_TOPA_PMI —** RTIT Table of Physical Addresses PMI. Allows software to differentiate between legacy PMI sources. |
| 54:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | WO | **CLEAR_FIXED_PERFMON_COUNTER_2_OVERFLOW —** Set to 1 to Clear OVF_FIXED_CTR2 bit. |
| 33 | Thread | 0h | WO | **CLEAR_FIXED_PERFMON_COUNTER_1_OVERFLOW —** Set to 1 to Clear OVF_FIXED_CTR1 bit. |
| 32 | Thread | 0h | WO | **CLEAR_FIXED_PERFMON_COUNTER_0_OVERFLOW —** Set to 1 to Clear OVF_FIXED_CTR0 bit. |
| 31:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | WO | **CLEAR_PERFMON_COUNTER_3_OVERFLOW —** Set to 1 to Clear OVF_PMC3 bit. |
| 2 | Thread | 0h | WO | **CLEAR_PERFMON_COUNTER_2_OVERFLOW —** Set to 1 to Clear OVF_PMC2 bit. |
| 1 | Thread | 0h | WO | **CLEAR_PERFMON_COUNTER_1_OVERFLOW —** Set to 1 to Clear OVF_PMC1 bit. |
| 0 | Thread | 0h | WO | **CLEAR_PERFMON_COUNTER_0_OVERFLOW —** Set to 1 to Clear OVF_PMC0 bit. |

## 69.2.95 (391h) IA32_PERF_GLOBAL_STATUS_SET

This register is used to enumerate the availability of PerfMon resources for use.

| MSR Address: 391h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Thread | 0h | WO | **SET_COND_CHNGD —** Sets IA32_PERF_GLOBAL_STATUS[63]. |
| 62 | Thread | 0h | WO | **SET_PEBS_BUFFER_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[62]. |
| 61 | - | - | - | Reserved. |
| 60 | Thread | 0h | WO | **SET_ASCI —** Set the ASCI bit |
| 59 | Thread | 0h | WO | **SET_COUNTERS_FROZEN —** Sets IA32_PERF_GLOBAL_STATUS[59]. |
| 58 | Thread | 0h | WO | **SET_LBRS_FROZEN —** Sets IA32_PERF_GLOBAL_STATUS[58]. |
| 57:56 | - | - | - | Reserved. |
| 55 | Thread | 0h | WO | **SET_RTIT_TOPA_PMI —** Sets IA32_PERF_GLOBAL_STATUS[55]. |
| 54:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | WO | **SET_FIX2_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[34]. |
| 33 | Thread | 0h | WO | **SET_FIX1_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[33]. |
| 32 | Thread | 0h | WO | **SET_FIX0_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[32]. |
| 31:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | WO | **SET_CNTR3_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[3]. |
| 2 | Thread | 0h | WO | **SET_CNTR2_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[2]. |
| 1 | Thread | 0h | WO | **SET_CNTR1_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[1]. |
| 0 | Thread | 0h | WO | **SET_CNTR0_OVFLW —** Sets IA32_PERF_GLOBAL_STATUS[0]. |

## 69.2.96 (392h) IA32_PERF_GLOBAL_INUSE

This register is used to enumerate the availability of PerfMon resources for use.

| MSR Address: 392h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Thread | 0h | RO | **PMI_PROGRAMMED —** Set if any of the Fixed or GP counters have PMI Enabled. |
| 62 | Thread | 0h | RO | **PEBS_PROGRAMMED —** PEBS_ENABLE[0]. |
| 61:35 | - | - | - | Reserved. |
| 34 | Thread | 0h | RO | **IA32_FIXED_CTR2_PROGRAMMED —** BITWISE_OR(Fixed_Ctr_Ctrl[9:8]). |
| 33 | Thread | 0h | RO | **IA32_FIXED_CTR1_PROGRAMMED —** BITWISE_OR(Fixed_Ctr_Ctrl[5:4]). |
| 32 | Thread | 0h | RO | **IA32_FIXED_CTR0_PROGRAMMED —** BITWISE_OR(Fixed_Ctr_Ctrl[1:0]). |
| 31:4 | - | - | - | Reserved. |
| 3 | Thread | 0h | RO | **IA32_PMC3_PROGRAMMED —** BITWISE_OR(Event_Select3[7:0]). |
| 2 | Thread | 0h | RO | **IA32_PMC2_PROGRAMMED —** BITWISE_OR(Event_Select2[7:0]). |
| 1 | Thread | 0h | RO | **IA32_PMC1_PROGRAMMED —** BITWISE_OR(Event_Select1[7:0]). |
| 0 | Thread | 0h | RO | **IA32_PMC0_PROGRAMMED —** BITWISE_OR(Event_Select0[7:0]). |

## 69.2.97 (3F1h) IA32_PEBS_ENABLE

Precise Event Based Sampling (PEBS) controlling MSR.

| MSR Address: 3F1h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:1 | - | - | - | Reserved. |
| 0 | Thread | 0h | RW | **PEBS_ENABLE —** When set, enables Precise Event Based Sampling. |

## 69.2.98 (3F8h) MSR_PKG_C3_RESIDENCY

Time spent in the Package C-State. It is given in units compatible to P1 clock frequency Guaranteed / Maximum Core Non-Turbo Frequency. This time will be updated by the SoC PCODE only after the C-State exit the update of this register has lower priority than actually ensuring that the C-State exit occurs.

| MSR Address: 3F8h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:60 | - | - | - | Reserved. |
| 59:0 | Package | 0h | RW | **COUNTER —** Counter Value. |

## 69.2.99 (3F9h) MSR_PKG_C6_RESIDENCY

Time spent in the Package C-State. It is given in units compatible to P1 clock frequency Guaranteed / Maximum Core Non-Turbo Frequency. This time will be updated by the SoC PCODE only after the C-State exit the update of this register has lower priority than actually ensuring that the C-State exit occurs.

| MSR Address: 3F9h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:60 | - | - | - | Reserved. |
| 59:0 | Package | 0h | RW | **COUNTER —** Counter Value. |

## 69.2.100 (3FCh) MSR_CORE_C3_RESIDENCY

The MSR_CORE_C3_RESIDENCY exists to assist in the debug of OEM platform power issues. This timer is not architectural which means it is not for use by OS or BIOS; and not guaranteed to be on future products or in the same MSR locations if it is available.

| MSR Address: 3FCh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | 0h | RW | **RESIDENCY —** This value will be accumulated by the SoC. This is in the CTC time-domain. A RDMSR will be required to convert this using "P" to get into the TSC time-domain. |

## 69.2.101 (3FDh) MSR_CORE_C6_RESIDENCY

The MSR_CORE_C6_RESIDENCY exists to assist in the debug of OEM platform power issues. This timer is not architectural which means it is not for use by OS or BIOS; and not guaranteed to be on future products or in the same MSR locations if it is available.

| MSR Address: 3FDh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | 0h | RW | **RESIDENCY —** This value will be accumulated by the SoC. This is in the CTC time-domain. A RDMSR will be required to convert this using "P" to get into the TSC time-domain. |

## 69.2.102 (3FEh) MSR_CORE_C7_RESIDENCY

The MSR_CORE_C7_RESIDENCY exists to assist in the debug of OEM platform power issues. This timer is not architectural which means it is not for use by OS or BIOS; and not guaranteed to be on future products or in the same MSR locations if it is available.

| MSR Address: 3FEh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | - | - | **DATA —** Reserved for possible future use. |

## 69.2.103 (400h) IA32_MC0_CTL (BIU)

| MSR Address: 400h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Module | - | - | Reserved. |
| 7:0 | Core | 0h | RW | **EN_IFU_ERR_SIGALING -** Enable IFU MCA error signaling. When bit is set, IFU errors will result in assertion of CATERR#. Otherwise, CATERR# is not asserted. |

## 69.2.104 (401h) IA32_MC0_STATUS

MCA Status Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 401h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Module | 0h | RW | **VALID —** Valid |
| 62 | Module | 0h | RW | **OVERFLOW —** Overflow |
| 61 | Module | 0h | RW | **MCERR —** MCERR |
| 60 | Module | 0h | RW | **EN —** Reserved. |
| 59 | Module | - | - | Reserved. |
| 58 | Module | 0h | RW | **ADDRV —** Addrv |
| 57 | Module | 0h | RW | **PCC —** XQ_ERR |
| 56:38 | Module | - | - | Reserved. |
| 37:32 | Module | 0h | RW | **BGF_ERROR —** Data driven sticky bit |
| 31:30 | Module | - | - | Reserved. |
| 29 | Module | 0h | RW | **MCERR_INBOUND —** MCERR observed bit |
| 28 | Module | 0h | RW | **MCERR_OUTBOUND —** MCERR driven sticky bit |
| 27:25 | Module | 0h | RW | **ERROR_TYPE —** Error type is valid when ADDRV is set:<br>• 001 - Inbound Data Error<br>• 010 - Outbound Data Error<br>• 011 - Transaction Error |
| 24:18 | Module | 0h | RW | **IDI_OPCODE —** XQ error request |
| 17:12 | Module | - | - | Reserved. |
| 11 | Module | 0h | RW | **MCACOD_IDI —** MCA COD bit1 |
| 10 | Module | 0h | RW | **MCACOD_ROB —** MCA COD ROB |
| 9:6 | Module | - | - | Reserved. |
| 5:4 | Module | 0h | RW | **MCACOD_REQ —** MCA COD req bits |
| 3:2 | Module | - | - | Reserved. |
| 1:0 | Module | 0h | RW | **MCACOD —** MCA COD bits |

## 69.2.105 (402h) IA32_MC0_ADDR (BIU)

MCA Address Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 402h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:40 | Module | - | - | Reserved. |
| 39:0 | Module | 0h | RW | **BITS0TO35 —** Bits 0 to 39 for the MCA address register. |

## 69.2.106 (404h) IA32_MC1_CTL (L2 Cache)

| MSR Address: 404h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:2 | Module | - | - | Reserved. |
| 1 | Module | 0h | RW | **PIC —** Enables reporting of MCA errors to Hard Error. |
| 0 | Module | 0h | RW | **DATA —** Enables reporting of MCA errors to Hard Error. |

## 69.2.107 (405h) IA32_MC1_STATUS (L2 Cache)

L2 MCA Status Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 405h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Module | 0h | RW | **MC_VALID_BIT —** MC is valid due to any ECC error detected. |
| 62 | Module | 0h | RW | **MC_OVWR_BIT —** New ECC error overwriting old ECC error. |
| 61 | Module | 0h | RW | **MC_UC_BIT —** Uncorrectable ECC error detected. |
| 60 | Module | 0h | RW | **MC_EN_BIT —** Any ECC error detected. |
| 59 | Module | - | - | Reserved. |
| 58 | Module | 0h | RW | **MC_ADDRV_BIT —** Address/State/LRU had ECC error. |
| 57 | Module | 0h | RW | **MC_PCC_BIT —** Processor Context Corruption, always 1 with an uncorrectable error. |
| 56:55 | Module | - | - | Reserved. |
| 54 | Module | 0h | RW | **MC_YELLOW_BIT —** Indicates parts health. |
| 53 | Module | 0h | RW | **MC_GREEN_BIT —** Indicates parts health only when SINGLE_MCA_LOGGING_MODE_RANGE = 00. |
| 52 | Module | 0h | RW | **MC_ERROR_COUNT_OVERFLOW_BIT —** Counting filtered correctable MCA errors. |
| 51:38 | Module | 0h | RW | **MC_ERROR_COUNT —** Counting filtered correctable MCA errors. |
| 37:36 | Module | - | - | Reserved. |
| 35 | Module | 0h | RW | **MCFH —** MCA Fast hit error. |
| 34 | Module | 0h | RW | **MCAC —** MCA Core ID. |
| 33:31 | Module | - | - | Reserved. |
| 30 | Module | 0h | RW | **MCSBF —** Fuse disable sbf pass through. |
| 29:25 | Module | - | - | Reserved. |
| 24:23 | Module | 0h | RW | **MSCOD —** Model Specific error code, in addition to MCA code.<br>Array with error:<br>• 00 - Data<br>• 01 - Tag<br>• 10 - State |

| MSR Address: 405h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 22:20 | Module | - | - | Reserved. |
| 19:16 | Module | 0h | RW | **MCAW —** MCA Way. |
| 15:0 | Core | - | RWS-O | **MCACOD —** Machine Check Architecture Error Code<br>Refer to Intel® 64 and IA-32 Architectures Software Developer Manual, Volume 3A: System Programming Guide, Part 1.<br>Bit[15:11] can be any value between 0x0-0x1f.<br>Bit [8] will set to 0x1 when Cache Hierarchy Error.<br>For the lower 8 bits, the compound error code format: bRRRR TTLL<br>RRRR: Cache Hierarchy Error RRRR.<br>• b0000: Generic Error<br>• b0001: Generic Read<br>• b0010: Generic Write<br>• b0011: Data Read<br>• b0100: Data Write<br>• b0101: Instruction Fetch<br>• b0110: Prefetch<br>• b0111: Eviction<br>• b1000: Snoop<br>• b1001-B1111: Reserved RRRR<br>TT: Cache Hierarchy Error TT.<br>• b00: Instruction<br>• b01: Data<br>• b10: Generic TT<br>• b11: Reserved TT<br>LL: Cache Hierarchy Error LL<br>• b00: Level 0 (core L1 cache)<br>• b01: Level 1 (core L2 cache)<br>• b10: Level 2 (L3/LLC cache)<br>• b11: Generic LL<br>For Simple Error, bit[2:0]:<br>• b000: No Error<br>• b001: Unclassified Error<br>• b010: Microcode ROM Parity Error<br>• b011: External Error<br>• b100: FRC Error<br>• b101: Internal Parity Error<br>• b110-b111: Simple Error<br>Following are examples of supported codes:<br>• 0x0174 - Eviction Error (e.g. WBINVD)<br>• 0x0134 - L1 Data Read Error<br>• 0x0144 - L1 Data Store Error<br>• 0x0164 - L1 Data Prefetch Error<br>• 0x0184 - L1 Data Snoop Error |

## 69.2.108 (406h) IA32_MC1_ADDR (L2 Cache)

L2 MCA Address Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 406h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:40 | Module | - | - | Reserved. |
| 39:0 | Module | 0h | RW | **MCA_ADDR —** MCA Address MCA Control Register. |

## 69.2.109 (408h) IA32_MC2_CTL (FEC)

Reserved for future use.

| MSR Address: 408h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:4 | - | - | - | **RSVD_63_4** - Reserved |
| 3:1 | Core | 0h | RW | **EN_DTLB_ERR_SIGNALING** - Enable DTLB error reporting. When bit is set DTLB errors will result in assertion of CATERR#. Otherwise, CATERR# is not asserted. |
| 0 | Core | 0h | RW | **EN_DCU_ERR_SIGNALING** - Enable DCU error reporting. When bit is set DCU errors will result in assertion of CATERR#. Otherwise, CATERR# is not asserted. |

## 69.2.110 (409h) IA32_MC2_STATUS (FEC)

MCA Status Register.Only a write of all 0s is allowed by WRMSR microcode.

| MSR Address: 409h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Core | 0h | RW | **VAL —** Valid bit - Indicates the information in this register is for a valid error report. |
| 62 | Core | 0h | RW | **OVER —** Machine check overflow - Indicates that there was another error already written to this register which is still in the handler (i.e., the VAL bit was already set when this error occurred). |
| 61 | Core | 0h | RW | **UC —** Uncorrected Error - Signifies the error being reported went uncorrected. |
| 60 | Core | 0h | RW | **EN —** MCI_CTL register has the associated Error Enable bit set. |
| 59 | Core | - | - | Reserved. |
| 58 | Core | 0h | RW | **ADDRV —** MCI_ADDR register holds valid data. |
| 57 | Core | 0h | RW | **PCC —** Processor Context Corrupt - Indicates that the processor state may have been corrupted by this MCA event. |
| 56:53 | Core | - | - | Reserved. |
| 52:38 | Core | 0h | RW | **CORRECTED_ERROR_COUNT —** Indicates the number of corrected errors. |

| MSR Address: 409h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 37:32 | Core | - | - | Reserved. |
| 31:16 | Core | 0h | RW | **MSCOD** - Model Specific Error Code<br>• 0x0000 - Tag Parity Error<br>• 0x0001 - Data Parity Error<br>• 0x0010 - Tag Parity Error<br>• 0x0011 - Data Parity Error<br>• 0x0004 - Internal Error |
| 15:0 | Core | - | RWS-O | **MCACOD** - Machine Check Architecture Code<br>Refer to Intel® 64 and IA-32 Architectures Software Developer Manual, Volume 3A: System Programming Guide, Part 1.<br>Bit[15:11] can be any value between 0x0-0x1f.<br>Bit[8] will set to 0x1 when Cache Hierarchy Error.<br>For the lower 8 bits, the compound error code Format: bRRRR TTLL<br>RRRR: Cache Hierarchy Error RRRR.<br>• b0000: Generic Error<br>• b0001: Generic Read<br>• b0010: Generic Write<br>• b0011: Data Read<br>• b0100: Data Write<br>• b0101: Instruction Fetch<br>• b0110: Prefetch<br>• b0111: Eviction<br>• b1000: Snoop<br>• b1001-b1111: Reserved RRRR<br>TT: Cache Hierarchy Error TT.<br>• b00: Instruction<br>• b01: Data<br>• b10: Generic TT<br>• b11: Reserved TT<br>LL: Cache Hierarchy Error LL<br>• b00: Level 0 (core L1 cache)<br>• b01: Level 1 (core L2 cache)<br>• b10: Level 2 (L3/LLC cache)<br>• b11: Generic LL<br>For Simple Error, bit[2:0]:<br>• b000: No Error<br>• b001: Unclassified Error<br>• b010: Microcode ROM Parity Error<br>• b011: External Error<br>• b100: FRC Error<br>• b101: Internal Parity Error<br>• b110-b111: Simple Error<br>Following are examples of supported codes:<br>• 0x0174 - Eviction Error (e.g. WBINVD)<br>• 0x0134 - L1 Data Read Error<br>• 0x0144 - L1 Data Store Error<br>• 0x0164 - L1 Data Prefetch Error<br>• 0x0184 - L1 Data Snoop Error |

## 69.2.111 (40Ah) IA32_MC2_ADDR (FEC)

MCA Address Register. Only a write of all 0s is allowed by WRMSR microcode.

| MSR Address: 40Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Core | - | - | Reserved. |
| 47:0 | Core | 0h | RW | **ADDRESS —** Linear Address[47:0] for I$ or ITAG and Physical Address [39:0] for SNOOP ITAG PARITY ERROR. |

## 69.2.112 (40Ch) IA32_MC3_CTL (MEC)

MCA Control Register. MRM b1598713:Dont #GP on writing ones to the upper part of MCi_CTL because the software model always write one to the whole 64bit.

| MSR Address: 40Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:2 | Core | - | - | Reserved. |
| 1 | Core | 0h | RW | **L1_TAG_PERR_UNCORR —** Enable uncorrectable data cache errors. |
| 0 | Core | 0h | RW | **L1_DATA_PERR_UNCORR —** Enable uncorrectable tag array errors. |

## 69.2.113 (40Dh) IA32_MC3_STATUS (MEC)

MCA Status Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 40Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Core | 0h | RW | **VAL —** Valid bit for the bank. |
| 62 | Core | 0h | RW | **OVER —** Bank overflowed (you missed one). |
| 61 | Core | 0h | RW | **UC —** Error was uncorrectable. |
| 60 | Core | 0h | RW | **EN —** Error was enabled. |
| 59 | Core | - | - | Reserved. |
| 58 | Core | 0h | RW | **ADDRV —** Address is valid. |
| 57 | Core | 0h | RW | **PCC —** Processor context corrupt. |
| 56:21 | Core | - | - | Reserved. |
| 20:16 | Core | 0h | RW | **MODEL_SPECIFIC_ERROR_CODE —** Model specific error code. |
| 15:0 | Core | 0h | RW | **MCA_ERROR_CODE —** Machine check error code. |

## 69.2.114 (40Eh) IA32_MC3_ADDR (MEC)

MCA Address Register. WRMSR microcode

The SoC generates #GP on non-zero writes.

| MSR Address: 40Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:40 | Core | - | - | Reserved. |
| 39:0 | Core | 0h | RW | **ADDRESS —** The address in the address register. |

## 69.2.115 (410h) IA32_MC4_CTL (SA)

Machine Check Architecture Control MSR.

| MSR Address: 410h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **EE63** — Error Enable 63 |
| 62 | Package | 0h | RO | **EE62** — Error Enable 62 |
| 61 | Package | 0h | RO | **EE61** — Error Enable 61 |
| 60 | Package | 0h | RO | **EE60** — Error Enable 60 |
| 59 | Package | 0h | RO | **EE59** — Error Enable 59 |
| 58 | Package | 0h | RO | **EE58** — Error Enable 58 |
| 57 | Package | 0h | RO | **EE57** — Error Enable 57 |
| 56 | Package | 0h | RO | **EE56** — Error Enable 56 |
| 55 | Package | 0h | RO | **EE55** — Error Enable 55 |
| 54 | Package | 0h | RO | **EE54** — Error Enable 54 |
| 53 | Package | 0h | RO | **EE53** — Error Enable 53 |
| 52 | Package | 0h | RO | **EE52** — Error Enable 52 |
| 51 | Package | 0h | RO | **EE51** — Error Enable 51 |
| 50 | Package | 0h | RO | **EE50** — Error Enable 50 |
| 49 | Package | 0h | RO | **EE49** — Error Enable 49 |
| 48 | Package | 0h | RO | **EE48** — Error Enable 48 |
| 47 | Package | 0h | RO | **EE47** — Error Enable 47 |
| 46 | Package | 0h | RO | **EE46** — Error Enable 46 |
| 45 | Package | 0h | RO | **EE45** — Error Enable 45 |
| 44 | Package | 0h | RO | **EE44** — Error Enable 44 |
| 43 | Package | 0h | RO | **EE43** — Error Enable 43 |
| 42 | Package | 0h | RO | **EE42** — Error Enable 42 |
| 41 | Package | 0h | RO | **EE41** — Error Enable 41 |
| 40 | Package | 0h | RO | **EE40** — Error Enable 40 |
| 39 | Package | 0h | RO | **EE39** — Error Enable 39 |
| 38 | Package | 0h | RO | **EE38** — Error Enable 38 |
| 37 | Package | 0h | RO | **EE37** — Error Enable 37 |
| 36 | Package | 0h | RO | **EE36** — Error Enable 36 |
| 35 | Package | 0h | RO | **EE35** — Error Enable 35 |
| 34 | Package | 0h | RO | **EE34** — Error Enable 34 |
| 33 | Package | 0h | RO | **EE33** — Error Enable 33 |
| 32 | Package | 0h | RO | **EE32** — Error Enable 32 |
| 31 | Package | 0h | RO | **EE31** — Error Enable 31 |
| 30 | Package | 0h | RO | **EE30** — Error Enable 30 |
| 29 | Package | 0h | RO | **EE29** — Error Enable 29 |
| 28 | Package | 0h | RO | **EE28** — Error Enable 28 |
| 27 | Package | 0h | RO | **EE27** — Error Enable 27 |
| 26 | Package | 0h | RO | **EE26** — Error Enable 26 |

| MSR Address: 410h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 25 | Package | 0h | RO | **EE25 —** Error Enable 25 |
| 24 | Package | 0h | RO | **EE24 —** Error Enable 24 |
| 23 | Package | 0h | RO | **EE23 —** Error Enable 23 |
| 22 | Package | 0h | RO | **EE22 —** Error Enable 22 |
| 21 | Package | 0h | RO | **EE21 —** Error Enable 21 |
| 20 | Package | 0h | RO | **EE20 —** Error Enable 20 |
| 19 | Package | 0h | RO | **EE19 —** Error Enable 19 |
| 18 | Package | 0h | RO | **EE18 —** Error Enable 18 |
| 17 | Package | 0h | RO | **EE17 —** Error Enable 17 |
| 16 | Package | 0h | RO | **EE16 —** Error Enable 16 |
| 15 | Package | 0h | RO | **EE15 —** Error Enable 15 |
| 14 | Package | 0h | RO | **EE14 —** Error Enable 14 |
| 13 | Package | 0h | RO | **EE13 —** Error Enable 13 |
| 12 | Package | 0h | RO | **EE12 —** Error Enable 12 |
| 11 | Package | 0h | RO | **EE11 —** Error Enable 11 |
| 10 | Package | 0h | RO | **EE10 —** Error Enable 10 |
| 9 | Package | 0h | RO | **EE09 —** Error Enable 09 |
| 8 | Package | 0h | RO | **EE08 —** Error Enable 08 |
| 7 | Package | 0h | RO | **EE07 —** Error Enable 07 |
| 6 | Package | 0h | RO | **XUCODE_ERR —** Error Enable 06 |
| 5 | Package | 0h | RO | **EE05 —** Error Enable 05 |
| 4 | Package | 0h | RO | **EE04 —** Error Enable 04 |
| 3 | Package | 0h | RO | **EE03 —** Error Enable 03 |
| 2 | Package | 0h | RO | **EE02 —** Error Enable 02 |
| 1 | Package | 0h | RO | **EE01 —** Error Enable 01 |
| 0 | Package | 0h | RO | **EE00 —** Error Enable 00 |

## 69.2.116 (411h) IA32_MC4_STATUS (SA)

Machine Check Architecture Status MSR.

| MSR Address: 411h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Core | 0h | RW | **VAL —** Valid bit - Indicates the information in this register is for a valid error report. |
| 62 | Core | 0h | RW | **OVER —** Machine check overflow - Indicates that there was another error already written to this register which is still in the handler (i.e., the VAL bit was already set when this error occurred). |
| 61 | Core | 0h | RW | **UC —** Uncorrected Error - Signifies the error being reported went uncorrected. |
| 60 | Core | 0h | RW | **EN —** MCI_CTL register has the associated Error Enable bit set. |
| 59 | Core | - | - | Reserved. |
| 58 | Core | 0h | RW | **ADDRV —** MCI_ADDR register holds valid data. |
| 57 | Core | 0h | RW | **PCC —** Processor Context Corrupt - Indicates that the processor state may have been corrupted by this MCA event. |
| 56:53 | Core | - | - | Reserved. |
| 52:38 | Core | 0h | RW | **CORRECTED_ERROR_COUNT —** Indicates the number of corrected errors. |

| MSR Address: 411h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 37:32 | Core | - | - | Reserved. |
| 31:16 | Core | 0h | RW | **MSCOD** - Model Specific Error Code<br>• 0x0000 - Tag Parity Error<br>• 0x0001 - Data Parity Error<br>• 0x0010 - Tag Parity Error<br>• 0x0011 - Data Parity Error<br>• 0x0004 - Internal Error |
| 15:0 | Core | - | RWS-O | **MCACOD** - Machine Check Architecture Code<br>Refer to Intel® 64 and IA-32 Architectures Software Developer Manual, Volume 3A: System Programming Guide, Part 1.<br>Bit[15:11] can be any value between 0x0-0x1f.<br>Bit[8] will set to 0x1 when Cache Hierarchy Error.<br>For the lower 8 bits, the compound error code Format: bRRRR TTLL<br>RRRR: Cache Hierarchy Error RRRR.<br>• b0000: Generic Error<br>• b0001: Generic Read<br>• b0010: Generic Write<br>• b0011: Data Read<br>• b0100: Data Write<br>• b0101: Instruction Fetch<br>• b0110: Prefetch<br>• b0111: Eviction<br>• b1000: Snoop<br>• b1001-b1111: Reserved RRRR<br>TT: Cache Hierarchy Error TT.<br>• b00: Instruction<br>• b01: Data<br>• b10: Generic TT<br>• b11: Reserved TT<br>LL: Cache Hierarchy Error LL<br>• b00: Level 0 (core L1 cache)<br>• b01: Level 1 (core L2 cache)<br>• b10: Level 2 (L3/LLC cache)<br>• b11: Generic LL<br>For Simple Error, bit[2:0]:<br>• b000: No Error<br>• b001: Unclassified Error<br>• b010: Microcode ROM Parity Error<br>• b011: External Error<br>• b100: FRC Error<br>• b101: Internal Parity Error<br>• b110-b111: Simple Error<br>Following are examples of supported codes:<br>• 0x0174 - Eviction Error (e.g. WBINVD)<br>• 0x0134 - L1 Data Read Error<br>• 0x0144 - L1 Data Store Error<br>• 0x0164 - L1 Data Prefetch Error<br>• 0x0184 - L1 Data Snoop Error |

## 69.2.117 (412h) IA32_MC4_ADDR (SA)

Machine Check Architecture Address MSR.

| MSR Address: 412h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RW | **ENH_MCA_AVAIL —** Available when enhanced MCA is in use. |

## 69.2.118 (413h) IA32_MC4_MISC (SA)

PCU Machine Check Architecture Misc MSR.

| MSR Address: 413h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:9 | Package | 0h | RW | **MODEL_SPECIFIC_INFO —** Available when enhanced MCA is in use. |
| 8:6 | Package | 0h | RW | **ADDRESS_MODE —** This field contains the address mode for the recorded address. The options are:<br>• 000 -> Segment Offset<br>• 001 -> Linear Address<br>• 010 -> Physical Address |
| 5:0 | Package | 0h | RW | **RECOVERABLE_ADDRESS_LSB —** This field contains the LSB of the recoverable error address. |

## 69.2.119 (414h) IA32_MC5_CTL (PMU)

Machine Check Architecture Control MSR.

| MSR Address: 414h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **EE63** — Error Enable 63 |
| 62 | Package | 0h | RO | **EE62** — Error Enable 62 |
| 61 | Package | 0h | RO | **EE61** — Error Enable 61 |
| 60 | Package | 0h | RO | **EE60** — Error Enable 60 |
| 59 | Package | 0h | RO | **EE59** — Error Enable 59 |
| 58 | Package | 0h | RO | **EE58** — Error Enable 58 |
| 57 | Package | 0h | RO | **EE57** — Error Enable 57 |
| 56 | Package | 0h | RO | **EE56** — Error Enable 56 |
| 55 | Package | 0h | RO | **EE55** — Error Enable 55 |
| 54 | Package | 0h | RO | **EE54** — Error Enable 54 |
| 53 | Package | 0h | RO | **EE53** — Error Enable 53 |
| 52 | Package | 0h | RO | **EE52** — Error Enable 52 |
| 51 | Package | 0h | RO | **EE51** — Error Enable 51 |
| 50 | Package | 0h | RO | **EE50** — Error Enable 50 |
| 49 | Package | 0h | RO | **EE49** — Error Enable 49 |
| 48 | Package | 0h | RO | **EE48** — Error Enable 48 |
| 47 | Package | 0h | RO | **EE47** — Error Enable 47 |
| 46 | Package | 0h | RO | **EE46** — Error Enable 46 |
| 45 | Package | 0h | RO | **EE45** — Error Enable 45 |
| 44 | Package | 0h | RO | **EE44** — Error Enable 44 |
| 43 | Package | 0h | RO | **EE43** — Error Enable 43 |
| 42 | Package | 0h | RO | **EE42** — Error Enable 42 |
| 41 | Package | 0h | RO | **EE41** — Error Enable 41 |
| 40 | Package | 0h | RO | **EE40** — Error Enable 40 |
| 39 | Package | 0h | RO | **EE39** — Error Enable 39 |
| 38 | Package | 0h | RO | **EE38** — Error Enable 38 |
| 37 | Package | 0h | RO | **EE37** — Error Enable 37 |
| 36 | Package | 0h | RO | **EE36** — Error Enable 36 |
| 35 | Package | 0h | RO | **EE35** — Error Enable 35 |
| 34 | Package | 0h | RO | **EE34** — Error Enable 34 |
| 33 | Package | 0h | RO | **EE33** — Error Enable 33 |
| 32 | Package | 0h | RO | **EE32** — Error Enable 32 |
| 31 | Package | 0h | RO | **EE31** — Error Enable 31 |
| 30 | Package | 0h | RO | **EE30** — Error Enable 30 |
| 29 | Package | 0h | RO | **EE29** — Error Enable 29 |
| 28 | Package | 0h | RO | **EE28** — Error Enable 28 |
| 27 | Package | 0h | RO | **EE27** — Error Enable 27 |
| 26 | Package | 0h | RO | **EE26** — Error Enable 26 |

| MSR Address: 414h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 25 | Package | 0h | RO | **EE25 —** Error Enable 25 |
| 24 | Package | 0h | RO | **EE24 —** Error Enable 24 |
| 23 | Package | 0h | RO | **EE23 —** Error Enable 23 |
| 22 | Package | 0h | RO | **EE22 —** Error Enable 22 |
| 21 | Package | 0h | RO | **EE21 —** Error Enable 21 |
| 20 | Package | 0h | RO | **EE20 —** Error Enable 20 |
| 19 | Package | 0h | RO | **EE19 —** Error Enable 19 |
| 18 | Package | 0h | RO | **EE18 —** Error Enable 18 |
| 17 | Package | 0h | RO | **EE17 —** Error Enable 17 |
| 16 | Package | 0h | RO | **EE16 —** Error Enable 16 |
| 15 | Package | 0h | RO | **EE15 —** Error Enable 15 |
| 14 | Package | 0h | RO | **EE14 —** Error Enable 14 |
| 13 | Package | 0h | RO | **EE13 —** Error Enable 13 |
| 12 | Package | 0h | RO | **EE12 —** Error Enable 12 |
| 11 | Package | 0h | RO | **EE11 —** Error Enable 11 |
| 10 | Package | 0h | RO | **EE10 —** Error Enable 10 |
| 9 | Package | 0h | RO | **EE09 —** Error Enable 09 |
| 8 | Package | 0h | RO | **EE08 —** Error Enable 08 |
| 7 | Package | 0h | RO | **EE07 —** Error Enable 07 |
| 6 | Package | 0h | RO | **EE06 —** Error Enable 06 |
| 5 | Package | 0h | RO | **EE05 —** Error Enable 05 |
| 4 | Package | 0h | RO | **EE04 —** Error Enable 04 |
| 3 | Package | 0h | RO | **EE03 —** Error Enable 03 |
| 2 | Package | 0h | RO | **EN_PCU_FW_ERR —** Error Enable 02 |
| 1 | Package | 0h | RO | **EN_PCU_UC_ERR —** Error Enable 01 |
| 0 | Package | 0h | RO | **EN_PCU_HW_ERR —** Error Enable 00 |

## 69.2.120 (415h) IA32_MC5_STATUS (PMU)

Machine Check Architecture Status MSR.

| MSR Address: 415h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW | **VAL —** MCi_STATUS Register Valid. |
| 62 | Package | 0h | RW | **OVER —** Error Overflow. |
| 61 | Package | 0h | RW | **UC —** Uncorrected Error. |
| 60 | Package | 0h | RW | **EN —** Error Reporting Enabled. |
| 59 | Package | 0h | RW | **MISCV —** MCi_MISC Register Valid. |
| 58 | Package | 0h | RW | **ADDRV —** MCi_ADDR Register Valid. |
| 57 | Package | 0h | RW | **PCC —** Processor Context Corrupted. |
| 56 | Package | 0h | RW | **S —** Signaling an uncorrected recoverable (UCR) error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 55 | Package | 0h | RW | **AR —** Recovery Action Required for UCR Error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 54:53 | Package | 0h | RW | **TES —** Threshold-Based Error Status - When MCG_CAP11 is not set, these bits are model specific. |
| 52:38 | Package | 0h | RW | **CORRECTED_ERROR_COUNT —** Corrected error count. |
| 37:32 | Package | 0h | RW | **OTHER_INFO —** Other information. |
| 31:16 | Package | 0h | RW | **MSCOD —** Model specific Error Code. |
| 15:0 | Package | 0h | RW | **MCACOD —** MCA Error Code. |

## 69.2.121 (416h) IA32_MC5_ADDR (PMU)

Machine Check Architecture Address MSR.

| MSR Address: 416h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RW | **ENH_MCA_AVAIL —** Available when enhanced MCA is in use. |

## 69.2.122 (417h) IA32_MC5_MISC (PMU)

PCU Machine Check Architecture Misc MSR.

| MSR Address: 417h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:9 | Package | 0h | RW | **MODEL_SPECIFIC_INFO —** Available when enhanced MCA is in use. |
| 8:6 | Package | 0h | RW | **ADDRESS_MODE —** This field contains the address mode for the recorded address. The options are:<br>• 000 -> Segment Offset<br>• 001 -> Linear Address<br>• 010 -> Physical Address |
| 5:0 | Package | 0h | RW | **RECOVERABLE_ADDRESS_LSB —** This field contains the LSB of the recoverable error address. |

## 69.2.123 (418h) IA32_MC6_CTL (SA)

Machine Check Architecture Control MSR.

| MSR Address: 418h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **EE63** — Error Enable 63 |
| 62 | Package | 0h | RO | **EE62** — Error Enable 62 |
| 61 | Package | 0h | RO | **EE61** — Error Enable 61 |
| 60 | Package | 0h | RO | **EE60** — Error Enable 60 |
| 59 | Package | 0h | RO | **EE59** — Error Enable 59 |
| 58 | Package | 0h | RO | **EE58** — Error Enable 58 |
| 57 | Package | 0h | RO | **EE57** — Error Enable 57 |
| 56 | Package | 0h | RO | **EE56** — Error Enable 56 |
| 55 | Package | 0h | RO | **EE55** — Error Enable 55 |
| 54 | Package | 0h | RO | **EE54** — Error Enable 54 |
| 53 | Package | 0h | RO | **EE53** — Error Enable 53 |
| 52 | Package | 0h | RO | **EE52** — Error Enable 52 |
| 51 | Package | 0h | RO | **EE51** — Error Enable 51 |
| 50 | Package | 0h | RO | **EE50** — Error Enable 50 |
| 49 | Package | 0h | RO | **EE49** — Error Enable 49 |
| 48 | Package | 0h | RO | **EE48** — Error Enable 48 |
| 47 | Package | 0h | RO | **EE47** — Error Enable 47 |
| 46 | Package | 0h | RO | **EE46** — Error Enable 46 |
| 45 | Package | 0h | RO | **EE45** — Error Enable 45 |
| 44 | Package | 0h | RO | **EE44** — Error Enable 44 |
| 43 | Package | 0h | RO | **EE43** — Error Enable 43 |
| 42 | Package | 0h | RO | **EE42** — Error Enable 42 |
| 41 | Package | 0h | RO | **EE41** — Error Enable 41 |
| 40 | Package | 0h | RO | **EE40** — Error Enable 40 |
| 39 | Package | 0h | RO | **EE39** — Error Enable 39 |
| 38 | Package | 0h | RO | **EE38** — Error Enable 38 |
| 37 | Package | 0h | RO | **EE37** — Error Enable 37 |
| 36 | Package | 0h | RO | **EE36** — Error Enable 36 |
| 35 | Package | 0h | RO | **EE35** — Error Enable 35 |
| 34 | Package | 0h | RO | **EE34** — Error Enable 34 |
| 33 | Package | 0h | RO | **EE33** — Error Enable 33 |
| 32 | Package | 0h | RO | **EE32** — Error Enable 32 |
| 31 | Package | 0h | RO | **EE31** — Error Enable 31 |
| 30 | Package | 0h | RO | **EE30** — Error Enable 30 |
| 29 | Package | 0h | RO | **EE29** — Error Enable 29 |
| 28 | Package | 0h | RO | **EE28** — Error Enable 28 |
| 27 | Package | 0h | RO | **EE27** — Error Enable 27 |
| 26 | Package | 0h | RO | **EE26** — Error Enable 26 |

| MSR Address: 418h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 25 | Package | 0h | RO | **EE25 —** Error Enable 25 |
| 24 | Package | 0h | RO | **EE24 —** Error Enable 24 |
| 23 | Package | 0h | RO | **EE23 —** Error Enable 23 |
| 22 | Package | 0h | RO | **EE22 —** Error Enable 22 |
| 21 | Package | 0h | RO | **EE21 —** Error Enable 21 |
| 20 | Package | 0h | RO | **EE20 —** Error Enable 20 |
| 19 | Package | 0h | RO | **EE19 —** Error Enable 19 |
| 18 | Package | 0h | RO | **EE18 —** Error Enable 18 |
| 17 | Package | 0h | RO | **EE17 —** Error Enable 17 |
| 16 | Package | 0h | RO | **EE16 —** Error Enable 16 |
| 15 | Package | 0h | RO | **EE15 —** Error Enable 15 |
| 14 | Package | 0h | RO | **EE14 —** Error Enable 14 |
| 13 | Package | 0h | RO | **EE13 —** Error Enable 13 |
| 12 | Package | 0h | RO | **EE12 —** Error Enable 12 |
| 11 | Package | 0h | RO | **EE11 —** Error Enable 11 |
| 10 | Package | 0h | RO | **EE10 —** Error Enable 10 |
| 9 | Package | 0h | RO | **EE09 —** Error Enable 09 |
| 8 | Package | 0h | RO | **EE08 —** Error Enable 08 |
| 7 | Package | 0h | RO | **EE07 —** Error Enable 07 |
| 6 | Package | 0h | RO | **EE06 —** Error Enable 06 |
| 5 | Package | 0h | RO | **EE05 —** Error Enable 05 |
| 4 | Package | 0h | RO | **EE04 —** Error Enable 04 |
| 3 | Package | 0h | RO | **EE03 —** Error Enable 03 |
| 2 | Package | 0h | RO | **EE02 —** Error Enable 02 |
| 1 | Package | 0h | RO | **EE01 —** Error Enable 01 |
| 0 | Package | 0h | RO | **EE00 —** Error Enable 00 |

## 69.2.124 (419h) IA32_MC6_STATUS (SA)

Machine Check Architecture Status MSR.

| MSR Address: 419h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW | **VAL —** MCi_STATUS Register Valid. |
| 62 | Package | 0h | RW | **OVER —** Error Overflow. |
| 61 | Package | 0h | RW | **UC —** Uncorrected Error. |
| 60 | Package | 0h | RW | **EN —** Error Reporting Enabled. |
| 59 | Package | 0h | RW | **MISCV —** MCi_MISC Register Valid. |
| 58 | Package | 0h | RW | **ADDRV —** MCi_ADDR Register Valid. |
| 57 | Package | 0h | RW | **PCC —** Processor Context Corrupted. |
| 56 | Package | 0h | RW | **S —** Signaling an uncorrected recoverable (UCR) error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 55 | Package | 0h | RW | **AR —** Recovery Action Required for UCR Error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 54:53 | Package | 0h | RW | **TES —** Threshold-Based Error Status - When MCG_CAP11 is not set, these bits are model specific. |
| 52:38 | Package | 0h | RW | **CORRECTED_ERROR_COUNT —** Corrected error count. |
| 37:32 | Package | 0h | RW | **OTHER_INFO —** Other information. |
| 31:16 | Package | 0h | RW | **MSCOD —** Model specific Error Code. |
| 15:0 | Package | 0h | RW | **MCACOD —** MCA Error Code. |

## 69.2.125 (41Ah) IA32_MC6_ADDR (SA)

Machine Check Architecture Address MSR.

| MSR Address: 41Ah | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Package | 0h | RW | **ENH_MCA_AVAIL —** Available when enhanced MCA is in use. |

## 69.2.126 (41Ch) IA32_MC7_CTL (MEM 0)

Machine Check Architecture Control MSR.

| MSR Address: 41Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **EE63** — Error Enable 63 |
| 62 | Package | 0h | RO | **EE62** — Error Enable 62 |
| 61 | Package | 0h | RO | **EE61** — Error Enable 61 |
| 60 | Package | 0h | RO | **EE60** — Error Enable 60 |
| 59 | Package | 0h | RO | **EE59** — Error Enable 59 |
| 58 | Package | 0h | RO | **EE58** — Error Enable 58 |
| 57 | Package | 0h | RO | **EE57** — Error Enable 57 |
| 56 | Package | 0h | RO | **EE56** — Error Enable 56 |
| 55 | Package | 0h | RO | **EE55** — Error Enable 55 |
| 54 | Package | 0h | RO | **EE54** — Error Enable 54 |
| 53 | Package | 0h | RO | **EE53** — Error Enable 53 |
| 52 | Package | 0h | RO | **EE52** — Error Enable 52 |
| 51 | Package | 0h | RO | **EE51** — Error Enable 51 |
| 50 | Package | 0h | RO | **EE50** — Error Enable 50 |
| 49 | Package | 0h | RO | **EE49** — Error Enable 49 |
| 48 | Package | 0h | RO | **EE48** — Error Enable 48 |
| 47 | Package | 0h | RO | **EE47** — Error Enable 47 |
| 46 | Package | 0h | RO | **EE46** — Error Enable 46 |
| 45 | Package | 0h | RO | **EE45** — Error Enable 45 |
| 44 | Package | 0h | RO | **EE44** — Error Enable 44 |
| 43 | Package | 0h | RO | **EE43** — Error Enable 43 |
| 42 | Package | 0h | RO | **EE42** — Error Enable 42 |
| 41 | Package | 0h | RO | **EE41** — Error Enable 41 |
| 40 | Package | 0h | RO | **EE40** — Error Enable 40 |
| 39 | Package | 0h | RO | **EE39** — Error Enable 39 |
| 38 | Package | 0h | RO | **EE38** — Error Enable 38 |
| 37 | Package | 0h | RO | **EE37** — Error Enable 37 |
| 36 | Package | 0h | RO | **EE36** — Error Enable 36 |
| 35 | Package | 0h | RO | **EE35** — Error Enable 35 |
| 34 | Package | 0h | RO | **EE34** — Error Enable 34 |
| 33 | Package | 0h | RO | **EE33** — Error Enable 33 |
| 32 | Package | 0h | RO | **EE32** — Error Enable 32 |
| 31 | Package | 0h | RO | **EE31** — Error Enable 31 |
| 30 | Package | 0h | RO | **EE30** — Error Enable 30 |
| 29 | Package | 0h | RO | **EE29** — Error Enable 29 |
| 28 | Package | 0h | RO | **EE28** — Error Enable 28 |
| 27 | Package | 0h | RO | **EE27** — Error Enable 27 |
| 26 | Package | 0h | RO | **EE26** — Error Enable 26 |

| MSR Address: 41Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 25 | Package | 0h | RO | **EE25 —** Error Enable 25 |
| 24 | Package | 0h | RO | **EE24 —** Error Enable 24 |
| 23 | Package | 0h | RO | **EE23 —** Error Enable 23 |
| 22 | Package | 0h | RO | **EE22 —** Error Enable 22 |
| 21 | Package | 0h | RO | **EE21 —** Error Enable 21 |
| 20 | Package | 0h | RO | **EE20 —** Error Enable 20 |
| 19 | Package | 0h | RO | **EE19 —** Error Enable 19 |
| 18 | Package | 0h | RO | **EE18 —** Error Enable 18 |
| 17 | Package | 0h | RO | **EE17 —** Error Enable 17 |
| 16 | Package | 0h | RO | **EE16 —** Error Enable 16 |
| 15 | Package | 0h | RO | **EE15 —** Error Enable 15 |
| 14 | Package | 0h | RO | **EE14 —** Error Enable 14 |
| 13 | Package | 0h | RO | **EE13 —** Error Enable 13 |
| 12 | Package | 0h | RO | **EE12 —** Error Enable 12 |
| 11 | Package | 0h | RO | **EE11 —** Error Enable 11 |
| 10 | Package | 0h | RO | **EE10 —** Error Enable 10 |
| 9 | Package | 0h | RO | **EE09 —** Error Enable 09 |
| 8 | Package | 0h | RO | **EE08 —** Error Enable 08 |
| 7 | Package | 0h | RO | **EE07 —** Error Enable 07 |
| 6 | Package | 0h | RO | **EE06 —** Error Enable 06 |
| 5 | Package | 0h | RO | **EE05 —** Error Enable 05 |
| 4 | Package | 0h | RO | **EE04 —** Error Enable 04 |
| 3 | Package | 0h | RO | **EE03 —** Error Enable 03 |
| 2 | Package | 0h | RO | **EE02 —** Error Enable 02 |
| 1 | Package | 0h | RO | **EE01 —** Error Enable 01 |
| 0 | Package | 0h | RO | **EE00 —** Error Enable 00 |

## 69.2.127 (41Dh) IA32_MC7_STATUS (MEM 0)

Machine Check Architecture Status MSR.

| MSR Address: 41Dh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW | **VAL —** MCi_STATUS Register Valid. |
| 62 | Package | 0h | RW | **OVER —** Error Overflow. |
| 61 | Package | 0h | RW | **UC —** Uncorrected Error. |
| 60 | Package | 0h | RW | **EN —** Error Reporting Enabled. |
| 59 | Package | 0h | RW | **MISCV —** MCi_MISC Register Valid. |
| 58 | Package | 0h | RW | **ADDRV —** MCi_ADDR Register Valid. |
| 57 | Package | 0h | RW | **PCC —** Processor Context Corrupted. |
| 56 | Package | 0h | RW | **S —** Signaling an uncorrected recoverable (UCR) error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 55 | Package | 0h | RW | **AR —** Recovery Action Required for UCR Error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 54:53 | Package | 0h | RW | **TES —** Threshold-Based Error Status - When MCG_CAP11 is not set, these bits are model specific. |
| 52:38 | Package | 0h | RW | **CORRECTED_ERROR_COUNT —** Corrected error count. |
| 37:32 | Package | 0h | RW | **OTHER_INFO —** Other information. |
| 31:16 | Package | 0h | RW-O | **MSCOD —** Model specific Error Code.<br>• 0x0200 - BT Timeout Error<br>• 0x0100 - Internal Parity<br>• 0x0080 - Internal Parity<br>• 0x0040 - Internal Parity<br>• 0x0020 - Internal Parity<br>• 0x0010 - RTID Error<br>• 0x0008 - Internal Error<br>• 0x0004 - Partial Write Error<br>• 0x0002 - Data Buffer Parity Error<br>• 0x0001 - Memory Read Error<br>• All others reserved. |
| 15:0 | Package | 0h | - | **MCCOD —** Specifies the machine check architecture defined error code for the machine check error condition detected.<br>Data Read Error: 0000 0000 1001 CCCC;<br>Generic Error: 0000 0000 1000 CCCC; (HADB Parity, RTID out Range, BGF, etc.)<br>Write Error: 0000 0000 1010 CCCC; (Partial Write, etc.)<br>Replace CCCC with channel number.<br>• Ch 0 = 0000b<br>• Ch 1 = 0001b<br>• Ch 2 = 0010b<br>• Ch 3 = 0011b<br>If the channel number can not be specified, such as request from Cbo and RTID out of range, use encode 1111. |

## 69.2.128 (41Fh) IA32_MC7_MISC (MEM 0)

PCU Machine Check Architecture Misc MSR.

| MSR Address: 41Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:9 | Package | 0h | RW | **MODEL_SPECIFIC_INFO —** Available when enhanced MCA is in use. |
| 8:6 | Package | 0h | RW | **ADDRESS_MODE —** This field contains the address mode for the recorded address. The options are:<br>• 000 -> Segment Offset<br>• 001 -> Linear Address<br>• 010 -> Physical Address |
| 5:0 | Package | 0h | RW | **RECOVERABLE_ADDRESS_LSB —** This field contains the LSB of the recoverable error address. |

## 69.2.129 (420h) IA32_MC8_CTL (MEM 1)

Machine Check Architecture Control MSR.

| MSR Address: 420h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **EE63** — Error Enable 63 |
| 62 | Package | 0h | RO | **EE62** — Error Enable 62 |
| 61 | Package | 0h | RO | **EE61** — Error Enable 61 |
| 60 | Package | 0h | RO | **EE60** — Error Enable 60 |
| 59 | Package | 0h | RO | **EE59** — Error Enable 59 |
| 58 | Package | 0h | RO | **EE58** — Error Enable 58 |
| 57 | Package | 0h | RO | **EE57** — Error Enable 57 |
| 56 | Package | 0h | RO | **EE56** — Error Enable 56 |
| 55 | Package | 0h | RO | **EE55** — Error Enable 55 |
| 54 | Package | 0h | RO | **EE54** — Error Enable 54 |
| 53 | Package | 0h | RO | **EE53** — Error Enable 53 |
| 52 | Package | 0h | RO | **EE52** — Error Enable 52 |
| 51 | Package | 0h | RO | **EE51** — Error Enable 51 |
| 50 | Package | 0h | RO | **EE50** — Error Enable 50 |
| 49 | Package | 0h | RO | **EE49** — Error Enable 49 |
| 48 | Package | 0h | RO | **EE48** — Error Enable 48 |
| 47 | Package | 0h | RO | **EE47** — Error Enable 47 |
| 46 | Package | 0h | RO | **EE46** — Error Enable 46 |
| 45 | Package | 0h | RO | **EE45** — Error Enable 45 |
| 44 | Package | 0h | RO | **EE44** — Error Enable 44 |
| 43 | Package | 0h | RO | **EE43** — Error Enable 43 |
| 42 | Package | 0h | RO | **EE42** — Error Enable 42 |
| 41 | Package | 0h | RO | **EE41** — Error Enable 41 |
| 40 | Package | 0h | RO | **EE40** — Error Enable 40 |
| 39 | Package | 0h | RO | **EE39** — Error Enable 39 |
| 38 | Package | 0h | RO | **EE38** — Error Enable 38 |
| 37 | Package | 0h | RO | **EE37** — Error Enable 37 |
| 36 | Package | 0h | RO | **EE36** — Error Enable 36 |
| 35 | Package | 0h | RO | **EE35** — Error Enable 35 |
| 34 | Package | 0h | RO | **EE34** — Error Enable 34 |
| 33 | Package | 0h | RO | **EE33** — Error Enable 33 |
| 32 | Package | 0h | RO | **EE32** — Error Enable 32 |
| 31 | Package | 0h | RO | **EE31** — Error Enable 31 |
| 30 | Package | 0h | RO | **EE30** — Error Enable 30 |
| 29 | Package | 0h | RO | **EE29** — Error Enable 29 |
| 28 | Package | 0h | RO | **EE28** — Error Enable 28 |
| 27 | Package | 0h | RO | **EE27** — Error Enable 27 |
| 26 | Package | 0h | RO | **EE26** — Error Enable 26 |

| MSR Address: 420h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 25 | Package | 0h | RO | **EE25 —** Error Enable 25 |
| 24 | Package | 0h | RO | **EE24 —** Error Enable 24 |
| 23 | Package | 0h | RO | **EE23 —** Error Enable 23 |
| 22 | Package | 0h | RO | **EE22 —** Error Enable 22 |
| 21 | Package | 0h | RO | **EE21 —** Error Enable 21 |
| 20 | Package | 0h | RO | **EE20 —** Error Enable 20 |
| 19 | Package | 0h | RO | **EE19 —** Error Enable 19 |
| 18 | Package | 0h | RO | **EE18 —** Error Enable 18 |
| 17 | Package | 0h | RO | **EE17 —** Error Enable 17 |
| 16 | Package | 0h | RO | **EE16 —** Error Enable 16 |
| 15 | Package | 0h | RO | **EE15 —** Error Enable 15 |
| 14 | Package | 0h | RO | **EE14 —** Error Enable 14 |
| 13 | Package | 0h | RO | **EE13 —** Error Enable 13 |
| 12 | Package | 0h | RO | **EE12 —** Error Enable 12 |
| 11 | Package | 0h | RO | **EE11 —** Error Enable 11 |
| 10 | Package | 0h | RO | **EE10 —** Error Enable 10 |
| 9 | Package | 0h | RO | **EE09 —** Error Enable 09 |
| 8 | Package | 0h | RO | **EE08 —** Error Enable 08 |
| 7 | Package | 0h | RO | **EE07 —** Error Enable 07 |
| 6 | Package | 0h | RO | **EE06 —** Error Enable 06 |
| 5 | Package | 0h | RO | **EE05 —** Error Enable 05 |
| 4 | Package | 0h | RO | **EE04 —** Error Enable 04 |
| 3 | Package | 0h | RO | **EE03 —** Error Enable 03 |
| 2 | Package | 0h | RO | **EE02 —** Error Enable 02 |
| 1 | Package | 0h | RO | **EE01 —** Error Enable 01 |
| 0 | Package | 0h | RO | **EE00 —** Error Enable 00 |

## 69.2.130 (421h) IA32_MC8_STATUS (MEM 1)

Machine Check Architecture Status MSR.

| MSR Address: 421h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW | **VAL —** MCi_STATUS Register Valid. |
| 62 | Package | 0h | RW | **OVER —** Error Overflow. |
| 61 | Package | 0h | RW | **UC —** Uncorrected Error. |
| 60 | Package | 0h | RW | **EN —** Error Reporting Enabled. |
| 59 | Package | 0h | RW | **MISCV —** MCi_MISC Register Valid. |
| 58 | Package | 0h | RW | **ADDRV —** MCi_ADDR Register Valid. |
| 57 | Package | 0h | RW | **PCC —** Processor Context Corrupted. |
| 56 | Package | 0h | RW | **S —** Signaling an uncorrected recoverable (UCR) error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 55 | Package | 0h | RW | **AR —** Recovery Action Required for UCR Error - When MCG_CAP11 or MCG_CAP24 are not set, these bits are reserved or model specific. |
| 54:53 | Package | 0h | RW | **TES —** Threshold-Based Error Status - When MCG_CAP11 is not set, these bits are model specific. |
| 52:38 | Package | 0h | RW | **CORRECTED_ERROR_COUNT —** Corrected error count. |
| 37:32 | Package | 0h | RW | **OTHER_INFO —** Other information. |
| 31:16 | Package | 0h | RW | **MSCOD —** Model specific Error Code. |
| 15:0 | Package | 0h | RW | **MCACOD —** MCA Error Code. |

## 69.2.131 (423h) IA32_MC8_MISC (MEM 1)

PCU Machine Check Architecture Misc MSR.

| MSR Address: 423h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:9 | Package | 0h | RW | **MODEL_SPECIFIC_INFO —** Available when enhanced MCA is in use. |
| 8:6 | Package | 0h | RW | **ADDRESS_MODE —** This field contains the address mode for the recorded address. The options are:<br>• 000 -> Segment Offset<br>• 001 -> Linear Address<br>• 010 -> Physical Address |
| 5:0 | Package | 0h | RW | **RECOVERABLE_ADDRESS_LSB —** This field contains the LSB of the recoverable error address. |

## 69.2.132 (4C1h) IA32_A_PMC0

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring.

| MSR Address: 4C1h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring. |

## 69.2.133 (4C2h) IA32_A_PMC1

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring.

| MSR Address: 4C2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring. |

## 69.2.134 (4C3h) IA32_A_PMC2

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring.

| MSR Address: 4C3h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring. |

## 69.2.135 (4C4h) IA32_A_PMC3

Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for complete details on MSRs related to Performance Monitoring.

| MSR Address: 4C4h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a complete details on MSRs related to Performance Monitoring. |

## 69.2.136 (4E0h) MSR_SMM_FEATURE_CONTROL

This MSR holds enable bits for on-chip storage of SMM and Code Access Check Violation features. This MSR can be written only from within SMM but can be read by SMM and non SMM code. This MSR is cleared only on cold reset.

| MSR Address: 4E0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:3 | Package | - | - | Reserved. |
| 2 | Package | 0h | RW | **SMM_CODE_CHK_EN —** If 1, generate unrecoverable MCHK on code fetch outside SMRR.<br>If 0, do not report any errors for code fetch outside SMRR. |
| 1 | Package | - | - | Reserved. |
| 0 | Package | 0h | RW | **LOCK —** If Lock = 1, CPU will drop writes to the MSR<br>If Lock = 0, CPU will allow update to b[2:0]. |

## 69.2.137 (4E2h) MSR_SMM_DELAYED

This MSR contains bit per logical processor indicating whether the logical processor is in the middle of long flow and hence will delay servicing of SMI. SMI will be serviced after the long flow completes execution.

| MSR Address: 4E2h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Package | - | - | Reserved. |
| 15:0 | Package | 0h | SMM-RO | **LOG_PROC_STATE —** Each bit represents a logical processor of its state in a long flow of internal operation which delays servicing an interrupt. The corresponding bit will be set at the start of long events such as:<br>- Microcode Update Load<br>- C6<br>- WBINVD<br>- Ratio Change<br>- Throttle<br>The bit is automatically cleared at the end of each long event. The reset value of this field is 0.<br>Only bit positions below N = CPUID.(EAX=0BH, ECX=PKG_LVL):EBX[15:0] can be updated. |

## 69.2.138 (4E3h) MSR_SMM_BLOCKED

This MSR contains bit per logical processor indicating whether the logical processor is in state where SMIs are blocked and hence will not be able to service SMI. SMI can be serviced after the logical processor exits the state in which SMIs are blocked.

| MSR Address: 4E3h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Package | - | - | Reserved. |
| 15:0 | Package | FFFFh | SMM-RO | **LOG_PROC —** Each bit represents a logical processor of its blocked state to service an SMI. The corresponding bit will be set if the logical processor is in one of the following states:<br>- Wait For SIPI<br>- SENTER Sleep<br>The reset value of this field is 0FFFH.<br>Only bit positions below N = CPUID.(EAX=0BH, ECX=PKG_LVL):EBX[15:0] can be updated. |

## 69.2.139 (500h) IA32_SGX_SVN_STATUS

| MSR Address: 500h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:24 | - | - | - | Reserved. |
| 23:16 | Package | 0h | RO | **SINIT_SE_SVN —** Since the SoC does not support SINIT, this bit field is always 0.<br>0 |
| 15:1 | - | - | - | Reserved. |
| 0 | Package | 0h | RO | **LOCK —** If 1, indicates that a non-faulting Intel® SGX instruction has been executed.<br>If 0, indicates that the processor will allow a properly signed Authenticated Code Modules (ACMs) to launch. |

## 69.2.140 (503h) MSR_SGX_DEBUG_MODE

This MSR does not survive warm reset.

| MSR Address: 503h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:2 | - | - | - | Reserved. |
| 1 | Package | 0h | RO | **NPK_ACTIVE —** This read-only field indicates that the Intel® Trace Hub is active. |
| 0 | Package | 0h | RW | **NPK_REQUEST —** When software sets this bit, a request is triggered to the Intel® Trace Hub to clear the Intel® Software Guard Extensions (Intel® SGX) keys. |

## 69.2.141 (560h) IA32_RTIT_OUTPUT_BASE

This MSR is used to configure the output region of internally-buffered packets. The size of the address field is determined by the maximum physical address width (MAXPHYADDR), as reported by CPUID.80000008H:EAX[7:0].

The processor updates this MSR while when packet generation is enabled, and those updates are asynchronous to instruction execution. Therefore, the values in this MSR should be considered unreliable unless packet generation is disabled (IA32_RTIT_CTL.TraceEn = 0).

This MSR can be written only when IA32_RTIT_CTL.TraceEn is 0; otherwise WRMSR causes a general-protection fault (#GP).

| MSR Address: 560h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for details on this register, including supported fields. |

## 69.2.142 (561h) IA32_RTIT_OUTPUT_MASK_PTRS

This MSR holds the pointers that indicate the ToPA entry that is currently in use and the offset into that entry's output region to which packets are being written.

The processor updates this MSR while when packet generation is enabled, and those updates are asynchronous to instruction execution. Therefore, the values in this MSR should be considered unreliable unless packet generation is disabled (IA32_RTIT_CTL.TraceEn = 0).

This MSR can be written only when IA32_RTIT_CTL.TraceEn is 0; otherwise WRMSR causes a general-protection fault (#GP).

| MSR Address: 561h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for details on this register, including supported fields. |

## 69.2.143 (570h) IA32_RTIT_CTL

The primary enable and control MSR for trace packet generation.

| MSR Address: 570h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for details on this register, including supported fields. |

## 69.2.144 (571h) ARR_CR_RTIT_STATUSIA32_RTIT_STATUS

The IA32_RTIT_STATUS MSR is readable and writable by software, but some bits (ContextEn, TriggerEn) are read-only and cannot be directly modified. The WRMSR instruction ignores these bits in the source operand (attempts to modify these bits are ignored and do not cause WRMSR to fault).

This MSR can only be written when IA32_RTIT_CTL.TraceEn is 0; otherwise WRMSR causes a general-protection fault (#GP). The processor does not modify the value of this MSR while TraceEn is 0 (software can modify it with WRMSR).

| MSR Address: 571h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for details on this register, including supported fields. |

## 69.2.145 (572h) IA32_RTIT_CR3_MATCH

CR3 matches IA32_RTIT_CR3_MATCH if the two registers are identical for bits 63:5; the lower 5 bits of CR3 and IA32_RTIT_CR3_MATCH are not compared.

This MSR can be written only when IA32_RTIT_CTL.TraceEn is 0; otherwise WRMSR causes a general-protection fault (#GP). IA32_RTIT_CR3_MATCH[4:0] are reserved and must be 0; an attempt to set those bits using WRMSR causes a #GP.

| MSR Address: 572h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Core | - | - | **DATA —** Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for details on this register, including supported fields. |

## 69.2.146 (580h) IA32_RTIT_ADDR0_A

Region ADDR0 start address. MSR used to support EIP filtering.

| MSR Address: 580h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Thread | 0h | RO | **IP0_SIGN_EXT —** On a creg read, bit 47 is sign extended into bits 63:48. This state is not stored. |
| 47:0 | Thread | 0h | RW | **IP0_ADDR —** Holds the base Linear Instruction Pointer (address) for the region defined as ADDR0. |

## 69.2.147 (581h) IA32_RTIT_ADDR0_B

Region ADDR0 end address. MSR used to support EIP filtering.

| MSR Address: 581h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Thread | 0h | RO | **IP0_SIGN_EXT —** On a read, bit 47 is sign extended into bits 63:48. This state is not stored. |
| 47:0 | Thread | 0h | RW | **IP0_ADDR —** Holds the limit Linear Instruction Pointer (address) for the region defined as ADDR0. |

## 69.2.148 (582h) IA32_RTIT_ADDR1_A

Region ADDR1 start address. MSR used to support EIP filtering.

| MSR Address: 582h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Thread | 0h | RO | **IP1_SIGN_EXT —** On a read, bit 47 is sign extended into bits 63:48. This state is not stored. |
| 47:0 | Thread | 0h | RW | **IP1_ADDR —** Holds the base Linear Instruction Pointer (address) for the region defined as ADDR1. |

## 69.2.149 (583h) IA32_RTIT_ADDR1_B

Region ADDR1 end address. MSR used to support EIP filtering.

| MSR Address: 583h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Thread | 0h | RO | **IP1_SIGN_EXT —** On a read, bit 47 is sign extended into bits 63:48. This state is not stored. |
| 47:0 | Thread | 0h | RW | **IP1_ADDR —** Holds the limit Linear Instruction Pointer (address) for the region defined as ADDR1. |

## 69.2.150 (600h) IA32_DS_AREA

Points to the linear address of the first byte of the DS buffer management area, which is used to manage the BTS and PEBS buffers. Refer to the IA-32 Intel® Architecture Software Developer's Manual for a detailed description and usage of this MSR and the Debug Store (DS) Mechanism.

| MSR Address: 600h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **DATA —** The linear address of the first byte of the DS buffer management area. Useful bits in this field depend on the address methodology in use when the register state is saved. |

## 69.2.151 (601h) MSR_VR_CURRENT_CONFIG

This register is used to limit the current consumption of the Primary power Plane. Primary Plane refers to the Vccin power rail. The processor will read this value during Reset Phase 4. On each slow loop, the maximum current possible will be calculated and the processor will send the appropriate PS Code according to the thresholds in this register. Set power state amp thresholds to 0A in order to disable that power state.

| MSR Address: 601h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:56 | - | - | - | Reserved. |
| 55:48 | Package | 0h | RW | **IUNIT_LL_RES —** Load line resistance in 1/2 mOhm steps for the IUNIT domain. IUNIT IccMax current will be multiplied by IUNIT_LL_RES resistance to calculate the voltage guardband necessary to manage for an IUNIT didt droop. |
| 47:40 | Package | 0h | RW | **GT_LL_RES —** Load line resistance in 1/2 mOhm steps for the GT domain. GT IccMax current will be multiplied by GT_LL_RES resistance to calculate the voltage guardband necessary to manage for GT didt droop. |
| 39:32 | Package | 0h | RW | **GLM_LL_RES —** Load line resistance in 1/2 mOhm steps for the GLM domain. GLM IccMax current will be multiplied by GLM_LL_RES resistance to calculate the voltage guardband necessary to manage for GLM didt droop. |
| 31:26 | - | - | - | Reserved. |
| 25:13 | Package | 0h | RW | **VCCGI_PS1_ICCMAX —** Peak current supported by the VR on the VGI rail when in PS1 mode. A value of 0 indicates VR support 0A of current in PS1 which will effectively disable PS1 use. Units given in PACKAGE_POWER_SKU_UNIT.CURRENT_UNIT. |
| 12:0 | Package | 0h | RW | **VCCGI_PS0_ICCMAX —** Peak current supported by the VR on the VGI rail. Assumes VR is in the highest power state possible (PS0). Units given in PACKAGE_POWER_SKU_UNIT.CURRENT_UNIT. |

## 69.2.152 (603h) MSR_VR_MISC_CONFIG

Input voltage regulator configuration parameters.

| MSR Address: 603h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:2 | - | - | - | Reserved. |
| 1 | Package | 1h | RW | **IDLE_ENTRY_RAMP_RATE —** Controls the VR ramp rate on package Cstate entry events. 1 fast 0 slow. If Decay is set it has priority. |
| 0 | Package | 1h | RW | **IDLE_EXIT_RAMP_RATE —** Controls the VR ramp rate on package Cstate wake events. 1 fast 0 slow. |

## 69.2.153 (606h) MSR_RAPL_POWER_UNIT

Defines units for calculating SKU power and timing parameters. PCODE will update the contents of this register.

| MSR Address: 606h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:28 | Package | 3h | RW | **RESISTANCE_UNIT —** Specifies the number of decimal bits in which current units in this register are specified.<br>0: steps of 1 Ohm<br>1: steps of 0.5 Ohms<br>2: steps of 0.25 Ohms etc... |
| 27:24 | Package | 3h | RW | **CURRENT_UNIT —** Specifies the number of decimal bits in which current units in this register are specified.<br>0: steps of 1 A<br>1: steps of 0.5 Amps<br>2: steps of 0.25 A etc... |
| 23:20 | Package | - | - | Reserved. |
| 19:16 | Package | Ah | RW | **TIME_UNIT —** Time Units used for power control registers. The actual unit value is calculated by 1 s / 2^(TIME_UNIT). The default value of Ah corresponds to 976 µs. |
| 15:13 | Package | - | - | Reserved. |
| 12:8 | Package | Eh | RW | **ENERGY_UNIT —** Energy Units used for power control registers. The actual unit value is calculated by 1 J / 2^(ENERGY_UNIT). The default value of 14 decimal corresponds to Ux.14 number. |
| 7:4 | Package | - | - | Reserved. |
| 3:0 | Package | 8h | RW | **PWR_UNIT —** Power Units used for power control registers. The actual unit value is calculated by 1 W / 2^(PWR_UNIT). The default value of 0110b corresponds to 1/64 W. |

## 69.2.154 (610h) MSR_PKG_POWER_LIMIT

The Package Power Limit Register allows platform BIOS to limit the power consumption of the processor to the specified values. POWER_LIMIT_1 and POWER_LIMIT_1_TIME are used to specify sustained long duration power limits and are around the TDP value of the processor. POWER_LIMIT_2 is used to specify the short duration power limit. The specified Power Limits are programmable only if bit 29 in PLATFORM_INFO (MSR CEh) is set. If this bit is not set then writes to this MSR will result in a GP fault.The Integrated Graphics driver CPM driver BIOS and OS can balance the power budget between the Primary Power Plane IA and the Secondary Power Plane GT via PRIMARY_PLANE_TURBO_POWER_LIMIT_MSR and SECONDARY_PLANE_TURBO_POWER_LIMIT_MSR.

| **MSR Address: 610h** | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW_L | **PKG_PWR_LIM_LOCK —** When set all settings in this register are locked and are treated as Read Only. This bit will typically set by BIOS during boot time or resume from Sx. |
| 62:56 | Package | - | - | Reserved. |
| 55:49 | Package | 0h | RW_L | **PKG_PWR_LIM_2_TIME** — This indicates the short duration time window over which Power_Limit_2 value should be maintained. This field is made up of two numbers from the following equation.<br>The processor power consumption will be controlled to the Power_Limit_2 value over an exponential weighted moving average of the time window.<br>Time Window = (float) ((1+(X/4)) * (2 ^ Y))<br>Where:<br>X = POWER_LIMIT_2_TIME[55:54]<br>Y = POWER_LIMIT_2_TIME[53:49]<br>The unit of measurement for this field is available from PACKAGE_POWER_SKU_UNIT (MSR 606h).<br>It is recommended that system BIOS not update this value during runtime. |
| 48 | Package | 0h | RW_L | **PKG_CLMP_LIM_2 —** When set, allows processor to go below the OS requested P States in order to maintain the power below Power_Limit_2 value specified below.<br>Both RAPL clamp bits CRITICAL_POWER_CLAMP_[2-1] are set to 1. |
| 47 | Package | 0h | RW_L | **PKG_PWR_LIM_2_EN —** When set, enables processor to apply control policies such that the package power does not exceed Power_Limit_2 value. BIOS must set Power_Limit_2_EN for proper turbo mode operation and set Power_Limit_2 field of this MSR to an appropriate value. |
| 46:32 | Package | 0h | RW_L | **PKG_PWR_LIM_2 —** Short Duration Power limit value which the package must not exceed. This field is specified in the units as identified by the PACKAGE_POWER_SKU_UNIT (MSR 606h).<br>Note: BIOS must enable Power_Limit_2 to enable Short Duration Power Limit for the processor. Recommended default value is 1.2 times the Long Duration Power Limit value (Power_Limit_1). |
| 31:24 | Package | - | - | Reserved. |
| 23:17 | Package | 0h | RW_L | **PKG_PWR_LIM_1_TIME —** This indicates the time window over which Power_Limit_1 value should be maintained. This field is made up of two numbers from the following equation.<br>The processor power consumption will be controlled to the Power_Limit_1 value over an exponential weighted moving average of the time window.<br>Time Window = (float) ((1+(X/4)) * (2 ^ Y))<br>Where:<br>X = POWER_LIMIT_1_TIME[23:22]<br>Y = POWER_LIMIT_1_TIME[21:17]<br>The unit of measurement for this field is available from PACKAGE_POWER_SKU_UNIT (MSR 606h). The maximum allowed value in this field is defined in PACKAGE_POWER_SKU (MSR 614h) PACKAGE_MAX_TIME field.<br>It is recommended that system BIOS not update this value during runtime. |

| MSR Address: 610h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 16 | Package | 0h | RW_L | **PKG_CLMP_LIM_1 —** When set, allows processor to go below the OS requested P States in order to maintain the power below Power_Limit_1 value specified below.<br>Both RAPL clamp bits CRITICAL_POWER_CLAMP_[2-1] are set to 1. |
| 15 | Package | 0h | RW_L | **PKG_PWR_LIM_1_EN —** When set, enables processor to apply control policies such that the package average power does not exceed Power_Limit_1 value over over an exponential weighted moving average of the time window. This bit is always enabled indicating Power_Limit_1 is always active. Because the processor must maintain the power consumption to the TDP value, Power_Limit_1 is always enabled. |
| 14:0 | Package | 0h | RW_L | **PKG_PWR_LIM_1 —** Average Power limit value which the package must not exceed over a time window as specified by Power_Limit_1_TIME field above.<br>The processor power consumption will be controlled to the Power_Limit_1 value over an exponential weighted moving average of the time window.<br>This field is specified in the units as identified by the PACKAGE_POWER_SKU_UNIT (MSR 606h). Initial value varies based on the SKU.<br>It is recommended that system BIOS not update this value during runtime. |

## 69.2.155 (614h) MSR_PKG_POWER_INFO

Defines allowed SKU power and timing parameters. PCODE will update the contents of this register.

| MSR Address: 614h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:55 | Package | - | - | Reserved. |
| 54:48 | Package | 12h | RW | **PKG_MAX_WIN —** The maximal time window allowed for the SoC product SKU. Higher values will be clamped to this value. The unit of measurement is defined in PACKAGE_POWER_SKU_UNIT_MSRTIME_UNIT. |
| 47 | Package | - | - | Reserved. |
| 46:32 | Package | 240h | RW | **PKG_MAX_PWR —** The maximal package power setting allowed for the SoC product SKU. Higher values will be clamped to this value. The maximum setting is typical not guaranteed. The default value for this field is determined by the SoC product SKU. The units for this value are defined in PACKAGE_POWER_SKU_MSRPWR_UNIT. |
| 31 | Package | - | - | Reserved. |
| 30:16 | Package | 60h | RW | **PKG_MIN_PWR —** The minimal package power setting allowed for the SoC product SKU. Lower values will be clamped to this value. The minimum setting is typical not guaranteed. The default value for this field is determined by the SoC product SKU. The units for this value are defined in PACKAGE_POWER_SKU_MSRPWR_UNIT. |
| 15 | Package | - | - | Reserved. |
| 14:0 | Package | 118h | RW | **PKG_TDP —** The package's Thermal Design Power (TDP) allowed for this part. The value is in the units identified in POWER_UNIT field in PACKAGE_POWER_SKU_UNIT (MSR 606h). |

## 69.2.156 (615h) PPL3_CONTROL

PL3 Control MSR.

| MSR Address: 615h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW_L | **LOCK —** Lock this MSR until next reset:<br>0 - unlocked<br>1 - locked |
| 62:48 | - | - | - | Reserved. |
| 47 | Package | 0h | RW_L | **PL4_ENABLE —** Power Limit 4 Enable:<br>0 => Algorithm disabled<br>1 => Algorithm enabled |
| 46:32 | Package | 0h | RW_L | **PMAX —** Power Limit 'PL4' or Pmax power limit in the units as described PACAKGE_POWER_SKU_UNIT MSR. |
| 31 | - | - | - | Reserved. |
| 30:24 | Package | 0h | RW_L | **DUTY_CYCLE —** Expressed in percentage (%). Clipped to a max value of 100%. |
| 23:17 | Package | 0h | RW_L | **TIME_WINDOW —** Duration over which duty cycle control will be maintained. (xxYYYY format) |
| 16 | - | - | - | Reserved. |
| 15 | Package | 0h | RW_L | **PL3_ENABLE —** Power Limit 3 Enable:<br>0 => Algorithm disabled<br>1 => Algorithm enabled |
| 14:0 | Package | 0h | RW_L | **POWER_LIMIT —** PL3 or PAppMax power level. A power reading above this will be interpreted as a violation (in increments of 1/8th Watt). |

## 69.2.157 (618h) MSR_DRAM_POWER_LIMIT

RAPL power limit for the DDR domain. This register is written by platform software and read by the SoC once per 1 ms. There are actually two instances of this register: one with MSR access and one with MMIO access. There is a separate PECI command which is also analogous.

| MSR Address: 618h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RW | **LOCKED —** When set this entire register becomes Read-Only. This bit will typically be set by BIOS during boot. |
| 62:24 | - | - | - | Reserved. |
| 23:22 | Package | 0h | RW | **LIMIT1_TIME_WINDOW_X —** Power Limit0 time window X value for DDR domain. Actual time_window for RAPL is: 1/1024 seconds 1x/4 2y. |
| 21:17 | Package | 0h | RW | **LIMIT1_TIME_WINDOW_Y —** Power Limit0 time window Y value for DDR domain. Actual time_window for RAPL is: 1/1024 seconds 1x/4 2y. |
| 16 | Package | - | - | Reserved. |
| 15 | Package | 0h | RW | **LIMIT1_ENABLE —** Power Limit0 enable bit for DDR domain. |
| 14:0 | Package | 0h | RW | **LIMIT1_POWER —** Power Limit0 for DDR domain. Units Watts Format11.3 Resolution0.125W Range02047.875W. |

## 69.2.158 (61Ch) MSR_DRAM_POWER_INFO

This register defines allowed DRAM power and timing parameters, and provides DRAM power range information for RAPL usage. It is recommended to program "DRAM_POWER_INFO" CSR and leave MSR_DRAM_POWER_INFO unlocked for OS software to modify.

| MSR Address: 61Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63 | Package | 0h | RO | **LOCK —** Lock bit to lock the Register. |
| 62:55 | - | - | - | Reserved. |
| 54:48 | Package | 28h | RO | **DRAM_MAX_WIN —** The maximal time window allowed for the DRAM. Higher values will be clamped to this value.<br>x = PKG_MAX_WIN[54:53]<br>y = PKG_MAX_WIN[52:48]<br>The timing interval window is Floating Point number given by 1.x*power(2,y).<br>The unit of measurement is defined in DRAM_POWER_INFO_UNIT_MSR[TIME_UNIT]. |
| 47 | - | - | - | Reserved. |
| 46:32 | Package | 258h | RO | **DRAM_MAX_PWR —** The maximal power setting allowed for DRAM. Higher values will be clamped to this value. The maximum setting is typical (not guaranteed).<br>The units for this value are defined in DRAM_POWER_INFO_UNIT_MSR[PWR_UNIT]. |
| 31 | - | - | - | Reserved. |
| 30:16 | Package | 78h | RO | **DRAM_MIN_PWR —** The minimal power setting allowed for DRAM. Lower values will be clamped to this value. The minimum setting is typical (not guaranteed).<br>The units for this value are defined in DRAM_POWER_INFO_UNIT_MSR[PWR_UNIT]. |
| 15 | - | - | - | Reserved. |
| 14:0 | Package | 118h | RO | **DRAM_TDP —** The Spec power allowed for DRAM The TDP setting is typical (not guaranteed).<br>The units for this value are defined in DRAM_POWER_INFO_UNIT_MSR[PWR_UNIT]. |

## 69.2.159 (633h) MSR_PKGC_IRTL3

The coordinated PackageAdditive above the core Interrupt Response Time is used for BIOS runtime control. This setting affects the selected package state. It is reflected to the IOS as part of the PM_REQ message for PMC usage as well. This register may be changed dynamically due to platform events AC/DC etc.The coordinated PackageAdditive above the core Interrupt Response Time is used for BIOS runtime control. This setting affects the selected package state. It is reflected to the IOS as part of the PM_REQ message for PMC usage as well. This register may be changed dynamically due to platform events AC/DC etc.

| MSR Address: 633h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Package | - | - | Reserved. |
| 15 | Package | 0h | RW | **VALID —** This field qualifies the validity of the Value field in this register. |
| 14:13 | Package | - | - | Reserved. |
| 12:10 | Package | 0h | RW | **MULTIPLIER —** This field indicates the unit of measurement that is defined for the Value field in this register. |
| 9:0 | Package | 0h | RW | **VALUE —** The Interrupt Response Time Limit is given in units defined in the Multiplier field of this register. |

## 69.2.160 (634h) MSR_PKGC_IRTL4

The coordinated PackageAdditive above the core Interrupt Response Time is used for BIOS runtime control. This setting affects the selected package state. It is reflected to the IOS as part of the PM_REQ message for PMC usage as well. This register may be changed dynamically due to platform events AC/DC etc.

| MSR Address: 634h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Package | - | - | Reserved. |
| 15 | Package | 0h | RW | **VALID —** This field qualifies the validity of the Value field in this register. |
| 14:13 | Package | - | - | Reserved. |
| 12:10 | Package | 0h | RW | **MULTIPLIER —** This field indicates the unit of measurement that is defined for the Value field in this register. |
| 9:0 | Package | 0h | RW | **VALUE —** The Interrupt Response Time Limit is given in units defined in the Multiplier field of this register. |

## 69.2.161 (635h) MSR_PKGC_IRTL5

The coordinated PackageAdditive above the core Interrupt Response Time is used for BIOS runtime control. This setting affects the selected package state. It is reflected to the IOS as part of the PM_REQ message for PMC usage as well. This register may be changed dynamically due to platform events AC/DC etc.

| MSR Address: 635h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:16 | Package | - | - | Reserved. |
| 15 | Package | 0h | RW | **VALID —** This field qualifies the validity of the Value field in this register. |
| 14:13 | Package | - | - | Reserved. |
| 12:10 | Package | 0h | RW | **MULTIPLIER —** This field indicates the unit of measurement that is defined for the Value field in this register. |
| 9:0 | Package | 0h | RW | **VALUE —** The Interrupt Response Time Limit is given in units defined in the Multiplier field of this register. |

## 69.2.162 (639h) MSR_PP0_ENERGY_STATUS

Reports total energy consumed. The counter will wrap around and continue counting when it reaches its limit. The energy status is reported in units which are defined in the ENERGY_UNIT field of MSR_RAPL_POWER_UNIT. Software will read this value and subtract the difference from last value read. The value of this register is updated every 1 ms.

| MSR Address: 639h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Package | - | - | Reserved. |
| 31:0 | Package | 0h | RO | **Total Energy Consumed —** The unsigned integer value represents the total amount of energy consumed since the last time this register was cleared. The unit of this field is specified by the ENERGY_UNIT field of MSR_RAPL_POWER_UNIT. |

## 69.2.163 (641h) MSR_PP1_ENERGY_STATUS

Reports total energy consumed. The counter will wrap around and continue counting when it reaches its limit. The energy status is reported in units which are defined in the ENERGY_UNIT field of MSR_RAPL_POWER_UNIT. Software will read this value and subtract the difference from last value read. The value of this register is updated every 1 ms.

| MSR Address: 641h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Package | - | - | Reserved. |
| 31:0 | Package | 0h | RO | **Total Energy Consumed —** The unsigned integer value represents the total amount of energy consumed since the last time this register was cleared. The unit of this field is specified by the ENERGY_UNIT field of MSR_RAPL_POWER_UNIT. |

## 69.2.164 (64Ch) MSR_TURBO_ACTIVATION_RATIO

This register is programmed by System BIOS to indicate the new maximum non-turbo ratio to be used by the processor. Attempts to write to this MSR when the PLATFORM_INFO[34:33] == 0 may result in a #GP fault.

| MSR Address: 64Ch | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Package | - | - | Reserved. |
| 31 | Package | 0h | RW_L | **TURBO_ACTIVATION_RATIO_LOCK —** When this bit is set it indicates that the contents of this register are locked. This bit is cleared only on a reset. |
| 30:8 | Package | - | - | Reserved. |
| 7:0 | Package | 0h | RW_L | **MAX_NON_TURBO_RATIO —** System BIOS can program this field to indicate the new Max non turbo ratio (in units of 100 MHz). All P state requests above this ratio are considered as Max Turbo ratio request.<br>Note: A value of 0 indicates that the feature is disabled. |

## 69.2.165 (64Fh) MSR_CORE_PERF_LIMIT_REASONS

Interface to allow software to determine what is causing resolved frequency to be clamped below the requested frequency.

| MSR Address: 64Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31 | Package | 0h | RW_0C | **QOS_LOG —** Set by the SoC and cleared by software. |
| 30 | Package | 0h | RW_0C | **MAX_EFFICIENCY_FREQ_LOG —** Set by the SoC and cleared by software. |
| 29 | Package | 0h | RW_0C | **MCT_LOG —** Set by the SoC and cleared by software. |
| 28 | Package | 0h | RW_0C | **EDP_LOG —** Set by the SoC and cleared by software. |
| 27 | Package | 0h | RW_0C | **MULTI_CORE_TURBO_LOG —** Set by the SoC and cleared by software. |
| 26 | Package | 0h | RW_0C | **VR_THERMALERT_LOG —** Set by the SoC and cleared by software. |
| 25 | Package | 0h | RW_0C | **IA_UTILIZATION_LOG —** Set by the SoC and cleared by software. |
| 24 | Package | 0h | RW_0C | **DEV3_LOG —** Set by the SoC and cleared by software. |
| 23 | Package | 0h | RW_0C | **DEV2_LOG —** Set by the SoC and cleared by software. |
| 22:20 | Package | - | - | Reserved. |
| 19 | Package | 0h | RW_0C | **PL2_LOG —** Set by the SoC and cleared by software. |
| 18 | Package | 0h | RW_0C | **PL1_LOG —** Set by the SoC and cleared by software. |
| 17 | Package | 0h | RW_0C | **THERMAL_LOG —** Set by the SoC and cleared by software. |
| 16 | Package | 0h | RW_0C | **PROCHOT_LOG —** Set by the SoC and cleared by software. |
| 15 | Package | 0h | RO | **QOS_STATUS —** Frequency is limited below the operating system or driver Quality-of-Service floor. |
| 14 | Package | 0h | RO | **MAX_EFFICIENCY_FREQ_STATUS —** Frequency is limited below the maximum efficiency frequency. |
| 13 | Package | 0h | RO | **MCT_STATUS —** Frequency is limited due to ratio change transition attenuation (MCT, prevents frequent ratio changes due to core C-state entry/exit). |
| 12 | Package | 0h | RO | **EDP_STATUS —** Frequency is limited due to a package-level EDP constraint. |
| 11 | Package | 0h | RO | **MULTI_CORE_TURBO_STATUS —** Frequency is limited due to effective multi-core turbo constraints, if applicable. |
| 10 | Package | 0h | RO | **VR_THERMALERT_STATUS — F**requency is limited due to a VR thermal excursion. |
| 9 | Package | 0h | RO | **IA_UTILIZATION_STATUS —** Frequency is limited due to autonomous utilization-based P-state control, if supported. |
| 8 | Package | 0h | RO | **DEV3_STATUS —** Frequency is limited due to Dev3 driver override. |
| 7 | Package | 0h | RO | **DEV2_STATUS —** Frequency is limited due to Dev2 driver override. |
| 6:4 | Package | - | - | Reserved. |
| 3 | Package | 0h | RO | **PL2_STATUS —** Frequency is limited due to a package-level PL2 excursion. |
| 2 | Package | 0h | RO | **PL1_STATUS —** Frequency is limited due to a package-level PL1 excursion. |
| 1 | Package | 0h | RO | **THERMAL_STATUS —** Frequency is limited due to thermal excursion. |
| 0 | Package | 0h | RO | **PROCHOT_STATUS —** Frequency is limited due to external PROCHOT assertion, if supported. |

## 69.2.166 (80Fh) IA32_X2APIC_SIVR

The spurious interrupt vector register has been extended for both legacy and extended xAPIC mode with a mechanism that enables software to perform directed EOIs to specific IOxAPICs in the system. Bit 12 of the spurious interrupt vector register controls the generation of the EOI broadcast. When this bit is set, broadcast EOIs will not be generated. Support for this feature can be detected by means of bit 24 in the Local xAPIC Version register. The feature is supported if this bit is set to 1.

| MSR Address: 80Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RW | **DIRECTED_EOI_EN —** This bit controls the generation of the EOI broadcast. When this bit is set, broadcast EOIs will not be generated. |
| 11:9 | Thread | - | - | Reserved. |
| 8 | Thread | 0h | RW | **SOFTWARE_ENABLE —**<br>• 1 = enabled<br>• 0 = disabled |
| 7:0 | Thread | 0h | RW | **SPURIOUS_VECTOR —** Vector number of the spurious vector. |

## 69.2.167 (828h) IA32_X2APIC_ESR

Extended xAPIC Error Status Register.

| MSR Address: 828h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Thread | - | - | Reserved. |
| 7 | Thread | 0h | RW | **ILLEGAL_REGISTER_ADDRESS —** APIC register access detected with an illegal address. |
| 6 | Thread | 0h | RW | **RECEIVE_ILLEGAL_VECTOR —** Interrupt received with an illegal vector. |
| 5 | Thread | 0h | RW | **SEND_ILLEGAL_VECTOR —** IPI with illegal vector attempted from ICR write. |
| 4 | Thread | 0h | Rsvd0 | **REDIRECTABLE_IPI —** IPI with illegal Low Priority delivery mode attempted from ICR write. |
| 3 | Thread | 0h | Rsvd0 | **RECEIVE_ACCEPT_ERROR —** Set when the local APIC detects that the message it received was not accepted by any APIC on the APIC bus, including itself. |
| 2 | Thread | 0h | Rsvd0 | **SEND_ACCEPT_ERROR —** Set when the local APIC detects that a message it sent was not accepted by any APIC on the APIC bus. |
| 1 | Thread | 0h | Rsvd0 | **RECEIVE_CHECKSUM_ERROR —** Set when the local APIC detects a checksum error for a message that it received on the APIC bus. |
| 0 | Thread | 0h | Rsvd0 | **SEND_CHECKSUM_ERROR —** Set when the local APIC detects a checksum error for a message that it sent on the APIC bus. |

## 69.2.168 (82Fh) IA32_X2APIC_LVT_CMCI

The Corrected Machine Check Interrupt LVT Register is an extension of the xAPIC in both legacy and extended xAPIC modes.

| MSR Address: 82Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 16 | Thread | 0h | RW | **MASK —** A value of 1 indicates that the interrupt is masked. |
| 15:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | - | - | Reserved. |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Correctable MCA Interrupt Vector. |

## 69.2.169 (830h) IA32_X2APIC_ICR

Extended xAPIC Interrupt Command Register.

| MSR Address: 830h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | 0h | RW | **DESTINATION_FIELD —** 32-bit destination ID of the IPI. |
| 31:20 | Thread | - | - | Reserved. |
| 19:18 | Thread | 0h | RW | **DESTINATION_SHORTHAND —** Destination Shorthand:<br>• 00b - No shorthand<br>• 01b - Self<br>• 10b - All including self<br>• 11b - All excluding self |
| 17:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | 0h | RW | **DESTINATION_MODE —** Destination Mode:<br>• 0b - Physical<br>• 1b - Logical |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** 4-K aligned interrupt vector address. |

## 69.2.170 (832h) IA32_X2APIC_LVT_TIMER

Extended xAPIC LVT Entry for the xAPIC Timer.

| MSR Address: 832h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:19 | Thread | - | - | Reserved. |
| 18:17 | Thread | 0h | RW | **TIMER_MODE —** Timer mode:<br>• 00b - One shot<br>• 01b - Periodic<br>• 10b - TSC Deadline<br>• 11b - Reserved |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11:8 | Thread | - | - | Reserved. |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Timer Interrupt Vector. |

## 69.2.171 (833h) IA32_X2APIC_LVT_THERMAL

Extended xAPIC LVT Entry for the Thermal Interrupt.

| MSR Address: 833h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | - | - | Reserved. |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Thermal Interrupt Vector. |

## 69.2.172 (834h) IA32_X2APIC_LVT_PMI

Extended xAPIC LVT Entry for PerfMon Interrupt.

| MSR Address: 834h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | - | - | Reserved. |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC PerfMon Interrupt Vector. |

## 69.2.173 (835h) IA32_X2APIC_LVT_LINT0

Extended xAPIC LVT Entry for Lint Pin 0 Interrupt.

| MSR Address: 835h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15 | Thread | 0h | RW | **TRIGGER_MODE —** Trigger Mode:<br>• 0b - Edge<br>• 1b - Level |
| 14 | Thread | 0h | RO | **REMOTE_IRR —** For fixed mode, level-triggered interrupts this flag is set when the local APIC accepts the interrupt for servicing and is reset when an EOI command is received from the processor. The meaning of this flag is undefined for edge-triggered interrupts and other delivery modes. |
| 13 | Thread | 0h | RW | **POLARITY —** 0: active high, 1: active low |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | - | - | Reserved. |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Lint Pin 0 Interrupt Vector. |

## 69.2.174 (836h) IA32_X2APIC_LVT_LINT1

Extended xAPIC LVT Entry for Lint Pin 0 Interrupt.

| MSR Address: 836h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 18:17 | Thread | 0h | Rsvd0 | **TIMER_MODE —** fill description here |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15 | Thread | 0h | RW | **TRIGGER_MODE —** Trigger Mode:<br>• 0b - Edge<br>• 1b - Level |
| 14 | Thread | 0h | RO | **REMOTE_IRR —** For fixed mode, level-triggered interrupts this flag is set when the local APIC accepts the interrupt for servicing and is reset when an EOI command is received from the processor. The meaning of this flag is undefined for edge-triggered interrupts and other delivery modes. |
| 13 | Thread | 0h | RW | **POLARITY —** 0: active high, 1: active low |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** Delivery Status:<br>• 0b - Idle<br>• 1b - Pending |
| 11 | Thread | - | - | Reserved. |
| 10:8 | Thread | 0h | RW | **DELIVERY_MODE —** Delivery Mode:<br>• 000b - Fixed priority<br>• 001b - Lowest Priority<br>• 010b - SMI<br>• 011b - Reserved<br>• 100b - NMI<br>• 101b - INIT<br>• 110b - Startup<br>• 111b - Reserved |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Lint Pin 1 Interrupt Vector. |

## 69.2.175 (837h) IA32_X2APIC_LVT_ERROR

Extended xAPIC LVT Entry for Error Interrupt.

| MSR Address: 837h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:17 | Thread | - | - | Reserved. |
| 16 | Thread | 0h | RW | **MASK —** Mask”<br>• 0b - Not masked<br>• 1b - Masked |
| 15:13 | Thread | - | - | Reserved. |
| 12 | Thread | 0h | RO | **DELIVERY_STATUS —** fill description here |
| 11:8 | Thread | - | - | Reserved. |
| 7:0 | Thread | 0h | RW | **VECTOR —** PIC Error Interrupt Vector. |

## 69.2.176 (838h) IA32_X2APIC_INIT_COUNT

Extended xAPIC Timer Initial Count register.

| MSR Address: 838h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | - | - | Reserved. |
| 31:0 | Thread | 0h | RW | **INIT_COUNT_REGISTER_VALUE —** APIC LVT Interrupt Timer Initial Count. |

## 69.2.177 (839h) IA32_X2APIC_CUR_COUNT

Extended xAPIC Timer Current Count register.

| MSR Address: 839h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | - | - | Reserved. |
| 31:0 | Thread | 0h | RW | **CURRENT_COUNT_REGISTER_VALUE —** APIC LVT Interrupt Timer Current Count. |

## 69.2.178 (83Eh) IA32_X2APIC_DIV_CONF

Extended xAPIC Timer Divide Config register.

| MSR Address: 83Eh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:4 | Thread | - | - | Reserved. |
| 3 | Thread | 0h | RW | **CFGCNT_HI —** This corresponds to the value in the Divide Config Register in the APIC. |
| 2 | Thread | - | - | Reserved. |
| 1:0 | Thread | 0h | RW | **CFGCNT_LO —** This corresponds to the value in the Divide Config Register in the APIC. |

## 69.2.179 (83Fh) IA32_X2APIC_SELF_IPI

Extended xAPIC mode provides a new fast self IPI interface. System software only specifies the vector associated with the interrupt.

| MSR Address: 83Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Thread | - | - | Reserved. |
| 7:0 | Thread | 0h | RW | **VECTOR —** Self IPI Vector. |

## 69.2.180 (C80h) IA32_DEBUG_FEATURE

Silicon Debug Feature Control.

| MSR Address: C80h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Package | - | - | Reserved. |
| 31 | Package | 0h | RO | **DEBUG_OCCURRED —** This "sticky bit" is set by hardware to indicate the status of bit 0. Default is 0. |
| 30 | Package | 0h | RW | **LOCK —** If 1, locks any further change to the MSR. The lock bit is set automatically on the first SMI assertion even if not explicitly set by BIOS. Default is 0. Once set, the lock remains until a cold reset. |
| 29:1 | - | - | - | Reserved. |
| 0 | Package | 0h | RW | **ENABLE —** BIOS sets this bit to 1 to enable Silicon Debug Features. Default is 0. |

## 69.2.181 (C8Fh) IA32_PQR_ASSOC

Resource Association Register.

| MSR Address: C8Fh | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:34 | Core | - | - | Reserved. |
| 33:32 | Core | 0h | RW | **COS —** COS Selector - core 0; It is programmed to specify which L2_QOS mask should apply to this logical processor. |
| 31:0 | Core | - | - | Reserved. |

## 69.2.182 (D10h) IA32_L2_QOS_MASK_0

L2 Class Of Service Mask - COS 0.

| MSR Address: D10h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Module | - | - | Reserved. |
| 7:0 | Module | 0h | RW | **WAY_MASK —** Bit vector of available L2 ways for COS 0 enforcement.<br>For a 512KB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 1 way.<br>For a 1 MB and 2 MB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 2 way. So bit 0 corresponds to ways 0, 1. Bit 1 corresponds to ways 2, 3. And so on. |

## 69.2.183 (D11h) IA32_L2_QOS_MASK_1

L2 Class Of Service Mask - COS 1.

| MSR Address: D11h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Module | - | - | Reserved. |
| 7:0 | Module | 0h | RW | **WAY_MASK —** Bit vector of available L2 ways for COS 1 enforcement.<br>For a 512KB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 1 way.<br>For a 1 MB and 2 MB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 2 way. So bit 0 corresponds to ways 0, 1. Bit 1 corresponds to ways 2, 3. And so on. |

## 69.2.184 (D12h) IA32_L2_QOS_MASK_2

L2 Class Of Service Mask - COS 2.

| MSR Address: D12h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Module | - | - | Reserved. |
| 7:0 | Module | 0h | RW | **WAY_MASK —** Bit vector of available L2 ways for COS 2 enforcement.<br>For a 512KB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 1 way.<br>For a 1 MB and 2 MB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 2 way. So bit 0 corresponds to ways 0, 1. Bit 1 corresponds to ways 2, 3. And so on. |

## 69.2.185 (D13h) IA32_L2_QOS_MASK_3

L2 Class Of Service Mask - COS 3.

| MSR Address: D13h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:8 | Module | - | - | Reserved. |
| 7:0 | Module | 0h | RW | **WAY_MASK —** Bit vector of available L2 ways for COS 3 enforcement.<br>For a 512KB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 1 way.<br>For a 1 MB and 2 MB L2 cache, each bit in IA32_L2_QOS_MASK_*[7:0] corresponds to 2 way. So bit 0 corresponds to ways 0, 1. Bit 1 corresponds to ways 2, 3. And so on. |

## 69.2.186 (D90h) IA32_BNDCFGS

Bound Configuration Supervisor State of Intel® Memory Protection Extensions (Intel® MPX) configuration.

| MSR Address: D90h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:12 | Thread | 0h | RW | **ADDRESS_BD —** Base Address of Bound Directory. |
| 11:2 | Thread | - | - | Reserved. |
| 1 | Thread | 0h | RW | **BNDPRESERVE —** Preserve the bounds registers for near branch instructions in the absence of the BND prefix. |
| 0 | Thread | 0h | RW | **ENABLE —** Enable Intel® MPX in supervisor mode. |

## 69.2.187 (DA0h) IA32_XSS

Extended Supervisor State Mask.

| MSR Address: DA0h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:9 | - | - | - | Reserved. |
| 8 | Thread | 0h | RW | **RTIT —** Trace Packet Configuration State. This bit controls whether XSAVES/ XRSTORS can save/load RTIT state. |
| 7:0 | - | - | - | Reserved. |

## 69.2.188 (C0000080h) IA32_EFER

Extended Feature Enable Register (EFER). This MSR may only be accessed when CPUID.(EAX=80000001h):EDX[20] or EDX[29] is set. Features of this register are to be enabled by the OS. BIOS may examine this register, but should not modify the contents of this register.

| MSR Address: C0000080h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 31:12 | Thread | - | - | Reserved. |
| 11 | Thread | 0h | RW | **NXE —** Execute Disable Bit Enable. When set enables the Execute Disable Bit capability. This feature is only usable when the Physical Address Extension paging mechanism is engaged.<br>The OS will control this bit, BIOS must not modify this bit. |
| 10 | Thread | 0h | RW | **LMA —** IA-32e Mode Active (Long Mode Active). When set, the processor is in IA-32e mode (Long Mode) and is either running in compatibility mode or 64- bit mode. If LMA = 0 (Default), the processor is running in legacy mode and behaves like a standard 32-bit IA-32 processor. |
| 9 | Thread | - | - | Reserved. |
| 8 | Thread | 0h | RW | **LME —** IA-32e Mode Enable (Long Mode Enable). When set enables the processor IA-32e mode operation.<br>IA-32e mode is not activated until PAE mode paging is enabled. The OS will control this bit, BIOS must not modify this bit. |
| 7:1 | Thread | - | - | Reserved. |
| 0 | Thread | 0h | RW | **SCE —** SYSCALL Enable. Enables SYSCALL/SYSRET instructions in 64-bit mode.<br>It is the responsibility of the OS to enable SYSCALL/SYSRET for 64-bit operation. The OS will control this bit, BIOS must not modify this bit. |

## 69.2.189 (C0000081h) IA32_STAR

System-Call Target Address Register (STAR). Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000081h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:48 | Thread | 0h | RW | **SYSRET_CS_SS —** SYSRET CS and SS Selector. |
| 47:32 | Thread | 0h | RW | **SYSCALL_CS_SS —** SYSCALL CS and SS Selector. |
| 31:0 | Thread | 0h | RW | **SYSCALL_LEGACY_EIP —** SYSCALL legacy EIP Selector. |

## 69.2.190 (C0000082h) IA32_LSTAR

IA-32e Mode System-Call Target Address Register. Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000082h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **TARGET_RIP —** Target RIP for 64-bit Mode Calling Program. |

## 69.2.191 (C0000083h) IA32_CSTAR

IA-32e Mode System-Call Target Address. Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000083h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **SYSCALL_TARGET_ADDR —** IA-32e Mode System Call Target Address. |

## 69.2.192 (C0000084h) IA32_FMASK

System-Call Flag Mask. Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000084h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | - | - | Reserved. |
| 31:0 | Thread | 0h | RW | **SYSCALL_FLAG_MASK —** System Call Flag Mask. If CPUID (EAX=80000001h):EDX[29] is set. |

### 69.2.193 (C0000100h) IA32_FS_BASE

FS base MSR. Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000100h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **FS_BASE —** Map of BASE Address of FS. If CPUID (EAX=80000001h):EDX[29] is set. |

### 69.2.194 (C0000101h) IA32_GS_BASE

GS base MSR. Refer to the Intel® 64 and IA-32 Architectures Software Developer's Manual for a detailed description and usage of this MSR.

| MSR Address: C0000101h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **GS_BASE —** Map of BASE Address of GS. If CPUID (EAX=80000001h):EDX[29] is set. |

### 69.2.195 (C0000102h) IA32_KERNEL_GS_BASE

Swap Target of BASE Address of GS.

| MSR Address: C0000102h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:0 | Thread | 0h | RW | **GS_BASE_SWAP_TARGET —** Swap Target of BASE Address of GS. |

### 69.2.196 (C0000103h) IA32_TSC_AUX

Auxiliary Time-Stamp Counter (TSC).

| MSR Address: C0000103h | | | | |
|---|---|---|---|---|
| **Bit** | **Scope** | **Default** | **Attribute** | **Description** |
| 63:32 | Thread | - | - | Reserved. |
| 31:0 | Thread | 0h | RW | **AUX —** Auxiliary signature of Time-Stamp Counter. |

§ §

# Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:

HW8076502444202S R381 HW8076502444301S R3L7 HW8076502639302S R388 HW8076502639402S R389 HW8076502639602S R38B HW8076502639802S R38D HW8076503693501S R3JX

LifeElectronics
Живое партнерство

**ООО «ЛайфЭлектроникс»** **“LifeElectronics” LLC**

*ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703*

*Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.*

*С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров*

*Мы предлагаем:*

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

*В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.*

*Конструкторский отдел помогает осуществить:*

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

*Тел: +7 (812) 336 43 04 (многоканальный)*
*Email: org@lifeelectronics.ru*

***www.lifeelectronics.ru***